Handout
DrawLine.c

```
/*
===============================================================================
 Name        : DrawLine.c
 Author      : $RJ
 Version     :
 Copyright   : $(copyright)
 Description : main definition
===============================================================================
*/
#include <cr_section_macros.h>
#include <NXP/crp.h>
#include "LPC17xx.h"                    /* LPC17xx definitions */
#include "ssp.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>

/* Be careful with the port number and location number, because
some of the location may not exist in that port. */
#define PORT_NUM            0

uint8_t src_addr[SSP_BUFSIZE];
uint8_t dest_addr[SSP_BUFSIZE];

#define ST7735_TFTWIDTH 127
#define ST7735_TFTHEIGHT 159

#define ST7735_CASET 0x2A
#define ST7735_RASET 0x2B
#define ST7735_RAMWR 0x2C
#define ST7735_SLPOUT 0x11
#define ST7735_DISPON 0x29

#define swap(x, y) {x = x + y; y = x - y; x = x - y ;}

// defining color values
#define LIGHTBLUE 0x00FFE0
#define GREEN 0x00FF00
#define DARKBLUE 0x000033
#define BLACK 0x000000
#define BLUE 0x0007FF
#define RED 0xFF0000
#define MAGENTA 0x00F81F
#define WHITE 0xFFFFFF
#define PURPLE 0xCC33FF
```

```
int _height = ST7735_TFTHEIGHT;
int _width = ST7735_TFTWIDTH;

void spiwrite(uint8_t c)
{

 int pnum = 0;
 src_addr[0] = c;
 SSP_SSELToggle( pnum, 0 );
 SSPSend( pnum, (uint8_t *)src_addr, 1 );
 SSP_SSELToggle( pnum, 1 );
}

void writecommand(uint8_t c)
{
 LPC_GPIO0->FIOCLR |= (0x1<<21);
 spiwrite(c);
}

void writedata(uint8_t c)
{
 LPC_GPIO0->FIOSET |= (0x1<<21);
 spiwrite(c);
}

void writeword(uint16_t c)
{
 uint8_t d;
 d = c >> 8;
 writedata(d);
 d = c & 0xFF;
 writedata(d);
}

void write888(uint32_t color, uint32_t repeat)
{
 uint8_t red, green, blue;
 int i;
 red = (color >> 16);
 green = (color >> 8) & 0xFF;
 blue = color & 0xFF;
 for (i = 0; i< repeat; i++) {
  writedata(red);
  writedata(green);
  writedata(blue);
 }

}
```

```
void setAddrWindow(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1)
{
 writecommand(ST7735_CASET);
 writeword(x0);
 writeword(x1);
 writecommand(ST7735_RASET);
 writeword(y0);
 writeword(y1);
}


void fillrect(int16_t x0, int16_t y0, int16_t x1, int16_t y1, uint32_t color)
{
 int16_t i;
 int16_t width, height;
 width = x1-x0+1;
 height = y1-y0+1;
 setAddrWindow(x0,y0,x1,y1);
 writecommand(ST7735_RAMWR);
 write888(color,width*height);
}

void lcddelay(int ms)
{
 int count = 24000;
 int i;

 for ( i = count*ms; i--; i > 0);
}

void lcd_init()
{
 int i;
 printf("LCD Demo Begins!!!\n");
 // Set pins P0.16, P0.21, P0.22 as output
 LPC_GPIO0->FIODIR |= (0x1<<16);

 LPC_GPIO0->FIODIR |= (0x1<<21);
 LPC_GPIO0->FIODIR |= (0x1<<22);

 // Hardware Reset Sequence
 LPC_GPIO0->FIOSET |= (0x1<<22);
 lcddelay(500);

 LPC_GPIO0->FIOCLR |= (0x1<<22);
 lcddelay(500);

 LPC_GPIO0->FIOSET |= (0x1<<22);
 lcddelay(500);
```

```
 // initialize buffers
 for ( i = 0; i < SSP_BUFSIZE; i++ )
 {
   src_addr[i] = 0;
   dest_addr[i] = 0;
 }

 // Take LCD display out of sleep mode
 writecommand(ST7735_SLPOUT);
 lcddelay(200);

 // Turn LCD display on
 writecommand(ST7735_DISPON);
 lcddelay(200);
}

void drawPixel(int16_t x, int16_t y, uint32_t color)
{
 if ((x < 0) || (x >= _width) || (y < 0) || (y >= _height))
 return;

 setAddrWindow(x, y, x + 1, y + 1);
 writecommand(ST7735_RAMWR);
 write888(color, 1);
}



/**************************************************************************
Descriptions:        Draw line function
parameters:          Starting point (x0,y0), Ending point(x1,y1) and color
Returned value:       None
**************************************************************************/

void drawLine(int16_t x0, int16_t y0, int16_t x1, int16_t y1, uint32_t color)
{
 int16_t slope = abs(y1 - y0) > abs(x1 - x0);

 if (slope) {
  swap(x0, y0);
  swap(x1, y1);
 }

 if (x0 > x1) {
  swap(x0, x1);
  swap(y0, y1);
 }
```

```
 int16_t dx, dy;
 dx = x1 - x0;
 dy = abs(y1 - y0);
 int16_t err = dx / 2;
 int16_t ystep;
 if (y0 < y1) {
  ystep = 1;
 }
 else {
  ystep = -1;
 }

 for (; x0 <= x1; x0++) {
  if (slope) {
   drawPixel(y0, x0, color);
  }
  else {
   drawPixel(x0, y0, color);
  }

  err -= dy;
  if (err < 0) {
   y0 += ystep;

   err += dx;
  }
 }
}

/*========================
 Main Function main()
========================*/

int main (void)
{
	uint32_t pnum = PORT_NUM;
	pnum = 0 ;

	if ( pnum == 0 )
		SSP0Init();
	else
		puts("Port number is not correct");
	lcd_init();
	fillrect(0, 0, ST7735_TFTWIDTH, ST7735_TFTHEIGHT, WHITE);
	int x0,x1,y0,y1;

	x0 = 20;
	x1 = 80;
	y0 = 60;
```

```
        y1 = 140;
        drawLine(x0,y0,x1,y1,PURPLE);
         return 0;
}
```