

```

/*
 * File:    main.c
 * Author:  Syed Tahmid Mahbub
 * Target PIC: PIC32MX250F128B
 */

#include "config.h"
#include "tft_master.h"
#include "tft_gfx.h"

/* Demo code for interfacing TFT (ILI9340 controller) to PIC32
 * The library has been modified from a similar Adafruit library
 * written for Arduino.
 * Below is the original text header from Adafruit, followed by the code
 */

/*****
This is an example sketch for the Adafruit 2.2" SPI display.
This library works with the Adafruit 2.2" TFT Breakout w/SD card
----> http://www.adafruit.com/products/1480

Check out the links above for our tutorials and wiring diagrams
These displays use SPI to communicate, 4 or 5 pins are required to
interface (RST is optional)
Adafruit invests time and resources providing this open source code,
please support Adafruit and open-source hardware by purchasing
products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries.
MIT license, all text above must be included in any redistribution
*****/

void testCircles(unsigned char radius, unsigned short color);
void testFastLines(unsigned short color1, unsigned short color2);
void testFillScreen(void);
void testFilledCircles(unsigned char radius, unsigned short color);
void testFilledRects(unsigned short color1, unsigned short color2);
void testFilledRoundRects(void);
void testRoundRects(void);
void testFilledTriangles(void);
void testLines(unsigned short color);
void testRects(unsigned short color);
void testRoundRects(void);
void testText(void);
void testTriangles(void);

```

```

char buffer[20];

void main(void) {
SYSTEMConfigPerformance(PBCLK);
unsigned char rotation;
ANSELA = 0; ANSELB = 0; CM1CON = 0; CM2CON = 0;
tft_init_hw();
tft_begin();

testFillScreen();
testText();
testLines(ILI9340_CYAN);
testFastLines(ILI9340_RED, ILI9340_BLUE);
testRects(ILI9340_GREEN);
testFilledRects(ILI9340_YELLOW, ILI9340_MAGENTA);
testFilledCircles(10, ILI9340_MAGENTA);
testCircles(10, ILI9340_WHITE);
testTriangles();
testFilledTriangles();
testRoundRects();
testFilledRoundRects();

while (1){
for(rotation=0; rotation<4; rotation++) {
tft_setRotation(rotation);
testText();
delay_ms(2000);
}
}

void testFillScreen() {
tft_fillScreen(ILI9340_BLACK);
tft_fillScreen(ILI9340_RED);
tft_fillScreen(ILI9340_GREEN);
tft_fillScreen(ILI9340_BLUE);
tft_fillScreen(ILI9340_BLACK);
}

void testText() {
tft_fillScreen(ILI9340_BLACK);
tft_setCursor(0, 0);
tft_setTextColor(ILI9340_WHITE); tft_setTextSize(1);

```

```

tft_writeString("Hello World!\n");
tft_setTextColor(ILI9340_YELLOW); tft_setTextSize(2);
sprintf(buffer,"%0.2f\n", 1234.56);
tft_writeString(buffer);
tft_setTextColor(ILI9340_RED); tft_setTextSize(3);
tft_writeString("DEADBEEF\n");
tft_setTextColor(ILI9340_GREEN);
tft_setTextSize(5);
tft_writeString("Groop\n");
tft_setTextSize(2);
tft_writeString("I implore thee\n");
tft_setTextSize(1);

tft_writeString("my foonting turlingdromes.\n");
tft_writeString("And hooptiously drangle me\n");
tft_writeString("with crinkly bindlewurdles,\n");
tft_writeString("Or I will rend thee\n");
tft_writeString("in the gobberwarts\n");
tft_writeString("with my blurglecruncheon,\n");
tft_writeString("see if I don't!\n");
}

void testLines(unsigned short color) {
    int    x1, y1, x2, y2,
           w = _width,
           h = _height;

    tft_fillScreen(ILI9340_BLACK);

    x1 = y1 = 0;
    y2 = h - 1;
    for(x2=0; x2<w; x2+=6) tft_drawLine(x1, y1, x2, y2, color);
    x2 = w - 1;
    for(y2=0; y2<h; y2+=6) tft_drawLine(x1, y1, x2, y2, color);

    tft_fillScreen(ILI9340_BLACK);

    x1 = w - 1;
    y1 = 0;
    y2 = h - 1;
    for(x2=0; x2<w; x2+=6) tft_drawLine(x1, y1, x2, y2, color);
    x2 = 0;
    for(y2=0; y2<h; y2+=6) tft_drawLine(x1, y1, x2, y2, color);

    tft_fillScreen(ILI9340_BLACK);

```

```

x1 = 0;
y1 = h - 1;
y2 = 0;
for(x2=0; x2<w; x2+=6) tft_drawLine(x1, y1, x2, y2, color);
x2 = w - 1;
for(y2=0; y2<h; y2+=6) tft_drawLine(x1, y1, x2, y2, color);

tft_fillScreen(ILI9340_BLACK);

x1 = w - 1;
y1 = h - 1;
y2 = 0;
for(x2=0; x2<w; x2+=6) tft_drawLine(x1, y1, x2, y2, color);
x2 = 0;
for(y2=0; y2<h; y2+=6) tft_drawLine(x1, y1, x2, y2, color);

}

void testFastLines(unsigned short color1, unsigned short color2) {
int x, y, w, h;
w = _width;
h = _height;

tft_fillScreen(ILI9340_BLACK);
for(y=0; y<h; y+=5) tft_drawFastHLine(0, y, w, color1);
for(x=0; x<w; x+=5) tft_drawFastVLine(x, 0, h, color2);

}

void testRects(unsigned short color) {
int n, i, i2,
    cx = _width / 2,
    cy = _height / 2;

tft_fillScreen(ILI9340_BLACK);
n = min(_width, _height);
for(i=2; i<n; i+=6) {
    i2 = i / 2;
    tft_drawRect(cx-i2, cy-i2, i, i, color);
}

}

void testFilledRects(unsigned short color1, unsigned short color2) {
int n, i, i2,
    cx = _width / 2 - 1,

```

```

        cy = _height / 2 - 1;

tft_fillScreen(ILI9340_BLACK);
n = min(_width, _height);
for(i=n; i>0; i-=6) {
    i2 = i / 2;
    tft_fillRect(cx-i2, cy-i2, i, i, color1);
    tft_drawRect(cx-i2, cy-i2, i, i, color2);
}
}

void testFilledCircles(unsigned char radius, unsigned short color) {
    int x, y, w = _width, h = _height, r2 = radius * 2;

    tft_fillScreen(ILI9340_BLACK);
    for(x=radius; x<w; x+=r2) {
        for(y=radius; y<h; y+=r2) {
            tft_fillCircle(x, y, radius, color);
        }
    }

    ////return micros() - start;
}

void testCircles(unsigned char radius, unsigned short color) {
    ////unsigned long start;
    int x, y, r2 = radius * 2,
        w = _width + radius,
        h = _height + radius;

    // Screen is not cleared for this one -- this is
    // intentional and does not affect the reported time.
    //start = micros();
    for(x=0; x<w; x+=r2) {
        for(y=0; y<h; y+=r2) {
            tft_drawCircle(x, y, radius, color);
        }
    }

    ////return micros() - start;
}

void testTriangles() {
    ////unsigned long start;
    int n, i, cx = _width / 2 - 1,

```

```

        cy = _height / 2 - 1;

tft_fillScreen(ILI9340_BLACK);
n = min(cx, cy);
//start = micros();
for(i=0; i<n; i+=5) {
    tft_drawTriangle(
        cx , cy - i, // peak
        cx - i, cy + i, // bottom left
        cx + i, cy + i, // bottom right
        tft_Color565(0, 0, i));
}

//return micros() - start;
}

void testFilledTriangles() {
//unsigned long start, t = 0;
int i, cx = _width / 2 - 1,
    cy = _height / 2 - 1;

tft_fillScreen(ILI9340_BLACK);
//start = micros();
for(i=min(cx,cy); i>10; i-=5) {
//start = micros();
tft_fillTriangle(cx, cy - i, cx - i, cy + i, cx + i, cy + i,
    tft_Color565(0, i, i));
//t += micros() - start;
tft_drawTriangle(cx, cy - i, cx - i, cy + i, cx + i, cy + i,
    tft_Color565(i, i, 0));
}

//return t;
}

void testRoundRects() {
//unsigned long start;
int w, i, i2,
    cx = _width / 2 - 1,
    cy = _height / 2 - 1;

tft_fillScreen(ILI9340_BLACK);
w = min(_width, _height);
//start = micros();
for(i=0; i<w; i+=6) {
    i2 = i / 2;

```

```
    tft_drawRoundRect(cx-i2, cy-i2, i, i, i/8, tft_Color565(i, 0, 0));
}
}

void testFilledRoundRects() {
    int i, i2, cx, cy;

    cx = _width / 2 - 1;
    cy = _height / 2 - 1;

    tft_fillScreen(ILI9340_BLACK);

    for(i=min(_width, _height); i>20; i-=6) {
        i2 = i / 2;
        tft_fillRoundRect(cx-i2, cy-i2, i, i, i/8, tft_Color565(0, i, 0));
    }
}
```