Name: .... Student ID: ....

Name: .... Student ID: ....

Laboratory 08
Interfacing to Graphic LCD

1. Introducing LCD

The liquid crystal is an organic compound which responds to an applied electric field by changing the alignment of its molecules, and hence the light polarisation which it introduces. A small quantity of liquid crystal is contained between two parallel glass plates. A suitable field can be applied if transparent electrodes are located on the glass surface. In conjunction with the external polarising light filters, light is either blocked, or transmitted by the display cell. The electrodes can be made in any pattern desired, including single digits or symbols. LCD consists of rod-shaped tiny molecules sandwiched between a flat piece of glass and an opaque substrate. These rod-shaped molecules in between the plates align into two different physical positions based on the electric charge applied to them. When electric charge is applied they align to block the light entering through them, whereas when no-charge is applied they become transparent. Light passing through makes the desired images appear.

ตัด

need to rewrite

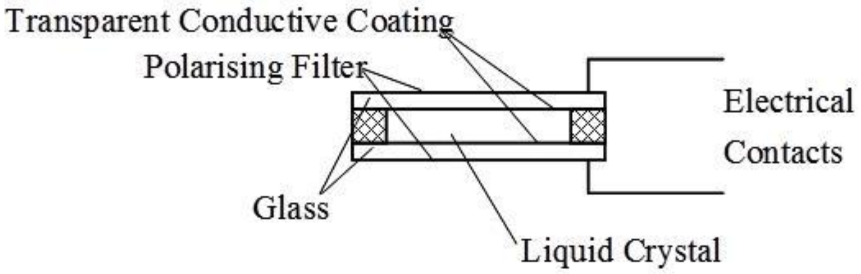

Fig.1. LCD Panel

An ubiquitous form of LCD is the character display. These are widely available from one line of characters to four or more, and are commonly seen on many domestic and office items. Driving this complex array of tiny LCD dots is far from simple, so such displays always contain a hidden microcontroller, customised to drive the display. LCD are connected to microcontroller through a LCD interface IC or directly to it's address and databus and few control pins, Few of the today's latest microcontrollers have built-in LCD driver. Also some of the LCDs support serial interface.

← พิ่มประเภทของ LCD ยกตัวอย่าง

2. ILI9341 TFT display driver library

The ILI9341 is a popular RGB colour display controller for use in small embedded systems. It is equipped with an on-chip frame buffer and TFT screen driver with voltage generator. It supports TFT screens with a size up to 240 x 320 pixels (WxH). It has 240 RGB pixels on scan line and 320 lines. Its RGB pixel resolution (bits per pixel) can be either 16-bit and 18-bit. It can use both parallel and serial interface.

Parallel bus size: 8-bit, 16-bit, 18 (32)-bit..

Addressing concept: Indexed bus interface (1 chip select pin + 1 address bit pin (Data/Command)),

Serial bus interfaces on ILI9341 by

SPI bus: 4-wire: /CS, DC, SDIN, SCLK.

SPI bus: 3-wire: /CS, SDIN, SCLK. SPI-3 uses a 9 bit byte frame where first bit is DC

The ILI9341 is supported by some driver library packages including the one used in this lab, the RAMTEX S6D0129. For more detail about its library you can find at: http://developer.mbed.org/users/dreschpe/code/SPI_TFT_ILI9341/docs/b2b3e5430f81/classSPI__TFT__ILI9341.html

3. mbed ILI9341 TFT display driver library

Public Member Functions

`SPI_TFT_ILI9341 (PinName mosi, PinName miso, PinName sclk, PinName cs, PinName reset, PinName dc, const char *name="TFT")`	Create a SPI_TFT object connected to SPI and three pins.
`virtual int width ()`	Get the width of the screen in pixel.
`virtual int height ()`	Get the height of the screen in pixel.
`virtual void pixel (int x, int y, int colour)`	Draw a pixel at x,y with color.
`void circle (int x, int y, int r, int colour)`	draw a circle
`void fillcircle (int x, int y, int r, int colour)`	draw a filled circle
`void line (int x0, int y0, int x1, int y1, int colour)`	draw a 1 pixel line
`void rect (int x0, int y0, int x1, int y1, int colour)`	draw a rect
`void fillrect (int x0, int y0, int x1, int y1, int colour)`	draw a filled rect
`virtual void locate (int x, int y)`	setup cursor position
`virtual void cls (void)`	Fill the screen with _backgroun color.
`virtual int columns (void)` *	calculate the max number of char in a line
`virtual int rows (void)` *	calculate the max number of columns
`virtual int _putc (int value)`	put a char on the screen
`virtual void character (int x, int y, int c)`	draw a character on given position out of the active font to the TFT

`void Bitmap (unsigned int x, unsigned int y, unsigned int w, unsigned int h, unsigned char *bitmap)`	paint a bitmap on the TFT
`int BMP_16 (unsigned int x, unsigned int y, const char *Name_BMP)`	paint a 16 bit BMP from filesytem on the TFT (slow)
`void set_font (unsigned char *f)`	select the font to use
`void set_orientation (unsigned int o)`	Set the orientation of the screen x,y: 0,0 is always top left.
`int Read_ID (void)`	read out the manufacturer ID of the LCD can used for checking the connection to the display
`virtual bool claim (FILE *stream)`	redirect output from a stream (stoud, sterr) to display

Protected Member Functions

`void WindowMax (void)`	Set draw window region to whole screen.
`void hline (int x0, int x1, int y, int colour)`	draw a horizontal line
`void vline (int y0, int y1, int x, int colour)`	draw a vertical line
`virtual void window (unsigned int x, unsigned int y, unsigned int w, unsigned int h)`	Set draw window region.
`void tft_reset ()`	Init the ILI9341 controller.
`void wr_dat (unsigned char value)`	Write data to the LCD controller.
`void wr_cmd (unsigned char value)`	Write a command the LCD controller.
`char rd_byte (unsigned char cmd)`	Start data sequence to the LCD controller.
`virtual void f_write (int data)`	Write a value to the to a LCD register.

4. Connecting the LCD module to the Xpresso board

In this lab we use 4-wire SPI interface to communicate with the LCD module. The pin connection are shown as follows

LCD Breakout	Xpresso pin
LCD MOSI	5 (initialized for SPI interface)
LCD MISO	6 (initialized for SPI interface)
LCD SCLK	7 (initialized for SPI interface)
LCD CS	8 (initialized for SPI interface)
LCD RESET	9 (initialized as Digital I/O)
LCD DC	10 (initialized as Digital I/O)
LCD LED	21 (initialized as Digital I/O)

5. Examples

```
#include <stdio.h>
#include "mbed.h"
#include "SPI_TFT_ILI9341.h"
#include "string"
#include "Arial12x12.h"
#include "Arial24x23.h"
#include "Arial28x28.h"
#include "font_big.h"

/*    extern unsigned char p1[];  // the mbed logo    */

DigitalOut LCD_LED(p21); // the Watterott display has a backlight switch

// the TFT is connected to SPI pin 5-7
SPI_TFT_ILI9341 TFT(p5, p6, p7, p8, p9, p10,"TFT"); // mosi, miso, sclk,
cs, reset, dc

int main()
{
    int i;
    LCD_LED = 1;  // backlight on

    TFT.claim(stdout);      // send stdout to the TFT display
    //TFT.claim(stderr);      // send stderr to the TFT display
    TFT.set_orientation(1);
    TFT.background(Black);    // set background to black
    TFT.foreground(White);    // set chars to white
    TFT.cls();                // clear the screen

    TFT.set_orientation(0);
    TFT.background(Black);
    TFT.cls();

    TFT.set_font((unsigned char*) Arial12x12);
    TFT.locate(0,0);
    printf("  Hello Mbed 0");
    TFT.set_orientation(1);
    TFT.locate(0,0);
    printf("  Hello Mbed 1");
    TFT.set_orientation(2);
    TFT.locate(0,0);
    printf("  Hello Mbed 2");
    TFT.set_orientation(3);
    TFT.locate(0,0);
    printf("  Hello Mbed 3");

    TFT.set_orientation(1);
    TFT.set_font((unsigned char*) Arial24x23);
    TFT.locate(50,100);
    TFT.printf("Kasin Vichienchom");
}
```

```
void drawGraphics() {
      // draw some graphics
          TFT.cls();
          TFT.set_font((unsigned char*) Arial24x23);
          TFT.locate(100,100);
          TFT.printf("Graphic");

          TFT.line(0,0,100,0,Green);
          TFT.line(0,0,0,200,Green);
          TFT.line(0,0,100,200,Green);

          TFT.rect(100,50,150,100,Red);
          TFT.fillrect(180,25,220,70,Blue);

          TFT.circle(80,150,33,White);
          TFT.fillcircle(160,190,20,Yellow);

          double s;

          for (i=0; i<320; i++) {
              s =20 * sin((long double) i / 10 );
              TFT.pixel(i,100 + (int)s ,Red);
          }
}

void BiggerText() {
  TFT.foreground(White);
  TFT.background(Blue);
  TFT.cls();
  TFT.set_font((unsigned char*) Arial24x23);
  TFT.locate(0,0);
  TFT.printf("Different Fonts :");

  TFT.set_font((unsigned char*) Neu42x35);
  TFT.locate(0,30);
  TFT.printf("Hello Mbed 1");
  TFT.set_font((unsigned char*) Arial24x23);
  TFT.locate(20,80);
  TFT.printf("Hello Mbed 2");
  TFT.set_font((unsigned char*) Arial12x12);
  TFT.locate(35,120);
  TFT.printf("Hello Mbed 3");
}
```

6. Experiment

1. Write a program to display your " Student ID Name Last name " on the LDC screen
2. Read current time from Xpresso board and display on the LCD screen
   Note that because our Xpresso does not have back up battery, therefore every time you disconnect USB cord and reconnect, it always starts at a default time. If you want it to show the correct time you must set it first.
3. Display an image of sinewave on the LCD screen

7. Reference

[1] Online document from mbed.org:
http://developer.mbed.org/users/dreschpe/code/SPI_TFT_ILI9341/docs/b2b3e5430f81/classSPI__TFT__ILI9341.html

[2] Rob Toulson and Tim Wilmshurst, Fast and Effective Embedded Systems Design, Elsevier, 2012