

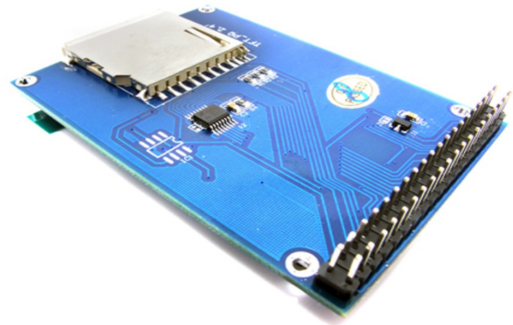
2.4' TFT LCD Screen Module with TOUCH

<http://www.elec freaks.com/store/24-tft-lcd-tft0124-p-110.html>

Bought in china in 2011; elder version of the display with ILI9325D controller; 3.3V version

This is a 3.3V LCD, so if you use it with a 5V platform like Arduino you'll need to convert pins to 3.3V

Test environment: arduino mega 2560 board (5V !)
powered by usb cable



When controlling the display with arduino mega, the interface signals carry 4.2V!!! This leads to latchup at databus signals. The display flickers and stays white at the end.

Solution (quick and dirty!): since the setup and write operation to the tft registers works, there is a quick and dirty solution: after tft controller is set up, the databus is disabled by switching the arduino pins to input thus not driving any tft controller pins.

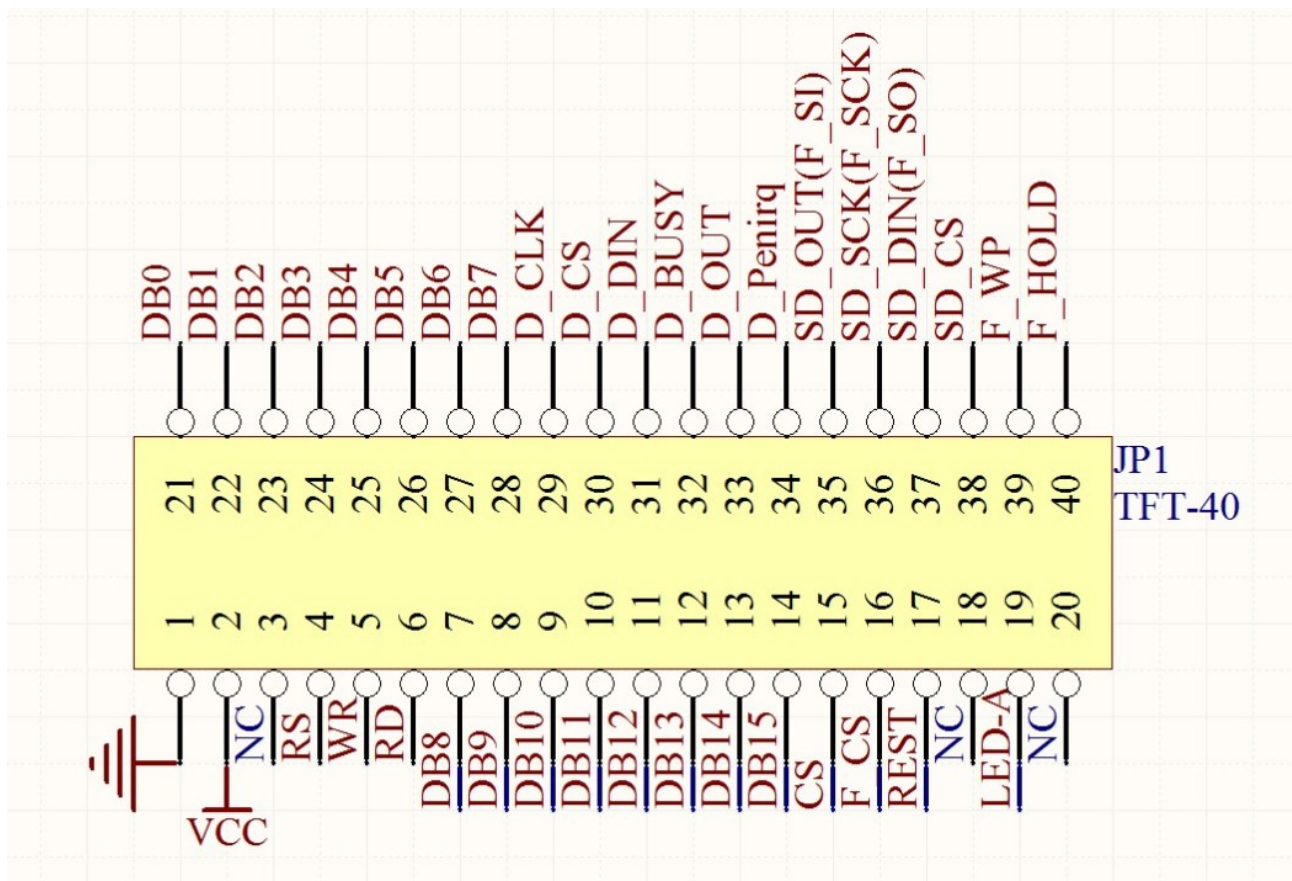
```
void deactivate_and_wait(){
  // set Databus inactive to prevent latchup
  for (int i = 22; i<=37; i++) pinMode(i,INPUT_PULLUP);
  delay(2000);
  for (int i = 22; i<=37; i++) pinMode(i,OUTPUT);
}
```

Solution (real solution) would be to use **3.3V interface** (e.g. level shifter or resistor dividers)

UTFT library and examples: ILI9325D_16 Controller interface (Attention: there are tft displays with other drivers e.g. S6D1121)

```
UTFT myGLCD(ILI9325D_16,38,39,40,41);
```

Default pinout for UTFT Examples



Signal	TFT Module pin	Arduino	
		2009 Uno Leonardo	Mega Due ³
DB0 ⁷	21	D8	D37
DB1 ⁷	22	D9	D36
DB2 ⁷	23	D10	D35
DB3 ⁷	24	D11	D34
DB4 ⁷	25	D12	D33
DB5 ⁷	26	D13	D32
DB6 ⁷	27	A0 (D14)	D31
DB7 ⁷	28	A1 (D15)	D30
DB8	7	D0	D22
DB9	8	D1	D23
DB10	9	D2	D24
DB11	10	D3	D25
DB12	11	D4	D26
DB13	12	D5	D27
DB14	13	D6	D28
DB15	14	D7	D29
RS	4		
WR	5		
RD	6		
CS	15		
REST	17		

Please note that these requirements are for the default configuration. If the hardware is not correctly configured, the requirements may be activated.

TFT ARDUINO

WR	5	39
CS	15	40
RESET	17	41
RD	6	3.3V

Example with dirty solution (may destroy your display!)

```
// UTFT_Demo_320x240
// Copyright (C)2015 Rinky-Dink Electronics, Henning Karlsen. All right reserved
// web: http://www.RinkyDinkElectronics.com/
//
// This program is a demo of how to use most of the functions
// of the library with a supported display modules.
//
// This demo was made for modules with a screen resolution
// of 320x240 pixels.
//
// This program requires the UTFT library.
//

#include <UTFT.h>

// Declare which fonts we will be using
extern uint8_t SmallFont[];

// Set the pins to the correct ones for your development shield
// -----
// Arduino Uno / 2009:
// -----
// Standard Arduino Uno/2009 shield          : <display model>,A5,A4,A3,A2
// DisplayModule Arduino Uno TFT shield      : <display model>,A5,A4,A3,A2
//
// Arduino Mega:
// -----
// Standard Arduino Mega/Due shield          : <display model>,38,39,40,41
// CTE TFT LCD/SD Shield for Arduino Mega    : <display model>,38,39,40,41
//
// Remember to change the model parameter to suit your display module!
UTFT myGLCD(5,38,39,40,41);

void setup()
{
    randomSeed(analogRead(0));

    // Setup the LCD
    myGLCD.InitLCD();
    myGLCD.setFont(SmallFont);
}

void loop()
{
    int buf[318];
    int x, x2;
    int y, y2;
    int r;

    // Clear the screen and draw the frame
    myGLCD.clrScr();
```

```

myGLCD.setColor(255, 0, 0);
myGLCD.fillRect(0, 0, 319, 13);
myGLCD.setColor(64, 64, 64);
myGLCD.fillRect(0, 226, 319, 239);
myGLCD.setColor(255, 255, 255);
myGLCD.setBackgroundColor(255, 0, 0);
myGLCD.print("* Universal Color TFT Display Library *", CENTER, 1);
myGLCD.setBackgroundColor(64, 64, 64);
myGLCD.setColor(255,255,0);
myGLCD.print("<http://www.RinkyDinkElectronics.com>", CENTER, 227);

myGLCD.setColor(0, 0, 255);
myGLCD.drawRect(0, 14, 319, 225);

// Draw crosshairs
myGLCD.setColor(0, 0, 255);
myGLCD.setBackgroundColor(0, 0, 0);
myGLCD.drawLine(159, 15, 159, 224);
myGLCD.drawLine(1, 119, 318, 119);
for (int i=9; i<310; i+=10)
    myGLCD.drawLine(i, 117, i, 121);
for (int i=19; i<220; i+=10)
    myGLCD.drawLine(157, i, 161, i);

// Draw sin-, cos- and tan-lines
myGLCD.setColor(0,255,255);
myGLCD.print("Sin", 5, 15);
for (int i=1; i<318; i++)
{
    myGLCD.drawPixel(i,119+(sin(((i*1.13)*3.14)/180)*95));
}

myGLCD.setColor(255,0,0);
myGLCD.print("Cos", 5, 27);
for (int i=1; i<318; i++)
{
    myGLCD.drawPixel(i,119+(cos(((i*1.13)*3.14)/180)*95));
}

myGLCD.setColor(255,255,0);
myGLCD.print("Tan", 5, 39);
for (int i=1; i<318; i++)
{
    myGLCD.drawPixel(i,119+(tan(((i*1.13)*3.14)/180)));
}

deactivate_and_wait();

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);
myGLCD.setColor(0, 0, 255);
myGLCD.setBackgroundColor(0, 0, 0);
myGLCD.drawLine(159, 15, 159, 224);
myGLCD.drawLine(1, 119, 318, 119);

// Draw a moving sinewave
x=1;
for (int i=1; i<(318*20); i++)
{
    x++;
    if (x==319)
        x=1;
}

```

```

    if (i>319)
    {
        if ((x==159)|| (buf[x-1]==119))
            myGLCD.setColor(0,0,255);
        else
            myGLCD.setColor(0,0,0);
        myGLCD.drawPixel(x,buf[x-1]);
    }
    myGLCD.setColor(0,255,255);
    y=119+(sin(((i*1.1)*3.14)/180)*(90-(i / 100)));
    myGLCD.drawPixel(x,y);
    buf[x-1]=y;
}

deactivate_and_wait();

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some filled rectangles
for (int i=1; i<6; i++)
{
    switch (i)
    {
        case 1:
            myGLCD.setColor(255,0,255);
            break;
        case 2:
            myGLCD.setColor(255,0,0);
            break;
        case 3:
            myGLCD.setColor(0,255,0);
            break;
        case 4:
            myGLCD.setColor(0,0,255);
            break;
        case 5:
            myGLCD.setColor(255,255,0);
            break;
    }
    myGLCD.fillRect(70+(i*20), 30+(i*20), 130+(i*20), 90+(i*20));
}

deactivate_and_wait();

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some filled, rounded rectangles
for (int i=1; i<6; i++)
{
    switch (i)
    {
        case 1:
            myGLCD.setColor(255,0,255);
            break;
        case 2:
            myGLCD.setColor(255,0,0);
            break;
        case 3:
            myGLCD.setColor(0,255,0);
            break;
    }
}

```

```

        case 4:
            myGLCD.setColor(0,0,255);
            break;
        case 5:
            myGLCD.setColor(255,255,0);
            break;
    }
    myGLCD.fillRect(190-(i*20), 30+(i*20), 250-(i*20), 90+(i*20));
}

deactivate_and_wait();

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some filled circles
for (int i=1; i<6; i++)
{
    switch (i)
    {
        case 1:
            myGLCD.setColor(255,0,255);
            break;
        case 2:
            myGLCD.setColor(255,0,0);
            break;
        case 3:
            myGLCD.setColor(0,255,0);
            break;
        case 4:
            myGLCD.setColor(0,0,255);
            break;
        case 5:
            myGLCD.setColor(255,255,0);
            break;
    }
    myGLCD.fillCircle(100+(i*20),60+(i*20), 30);
}

deactivate_and_wait();

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some lines in a pattern
myGLCD.setColor (255,0,0);
for (int i=15; i<224; i+=5)
{
    myGLCD.drawLine(1, i, (i*1.44)-10, 224);
}
myGLCD.setColor (255,0,0);
for (int i=224; i>15; i-=5)
{
    myGLCD.drawLine(318, i, (i*1.44)-11, 15);
}
myGLCD.setColor (0,255,255);
for (int i=224; i>15; i-=5)
{
    myGLCD.drawLine(1, i, 331-(i*1.44), 15);
}
myGLCD.setColor (0,255,255);
for (int i=15; i<224; i+=5)

```

```

{
    myGLCD.drawLine(318, i, 330-(i*1.44), 224);
}

deactivate_and_wait();

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some random circles
for (int i=0; i<100; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    x=32+random(256);
    y=45+random(146);
    r=random(30);
    myGLCD.drawCircle(x, y, r);
}

deactivate_and_wait();

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some random rectangles
for (int i=0; i<100; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    x=2+random(316);
    y=16+random(207);
    x2=2+random(316);
    y2=16+random(207);
    myGLCD.drawRect(x, y, x2, y2);
}

deactivate_and_wait();

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some random rounded rectangles
for (int i=0; i<100; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    x=2+random(316);
    y=16+random(207);
    x2=2+random(316);
    y2=16+random(207);
    myGLCD.drawRoundRect(x, y, x2, y2);
}

deactivate_and_wait();

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

for (int i=0; i<100; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    x=2+random(316);
    y=16+random(209);
    x2=2+random(316);

```

```

    y2=16+random(209);
    myGLCD.drawLine(x, y, x2, y2);
}

deactivate_and_wait();

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

for (int i=0; i<10000; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    myGLCD.drawPixel(2+random(316), 16+random(209));
}

deactivate_and_wait();

myGLCD.fillScr(0, 0, 255);
myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect(80, 70, 239, 169);

myGLCD.setColor(255, 255, 255);
myGLCD.setBackgroundColor(255, 0, 0);
myGLCD.print("That's it!", CENTER, 93);
myGLCD.print("Restarting in a", CENTER, 119);
myGLCD.print("few seconds...", CENTER, 132);

myGLCD.setColor(0, 255, 0);
myGLCD.setBackgroundColor(0, 0, 255);
myGLCD.print("Runtime: (msecs)", CENTER, 210);
myGLCD.printNumI(millis(), CENTER, 225);

deactivate_and_wait();
}

void deactivate_and_wait(){
    // set Databus inactive to prevent latchup
    for (int i = 22; i<=37; i++) pinMode(i,INPUT_PULLUP);
    delay(2000);
    for (int i = 22; i<=37; i++) pinMode(i,OUTPUT);
}

```

Touch

utouch Library

//tclk, byte tcs, byte tdin, byte dout, byte irq

UTouch myTouch(6, 5, 4, 3, 2);

Connections:

Name	Display	Arduino Mega
clk	29	6
cs	30	5
din	31	4

dout	33	3
irq	34	2