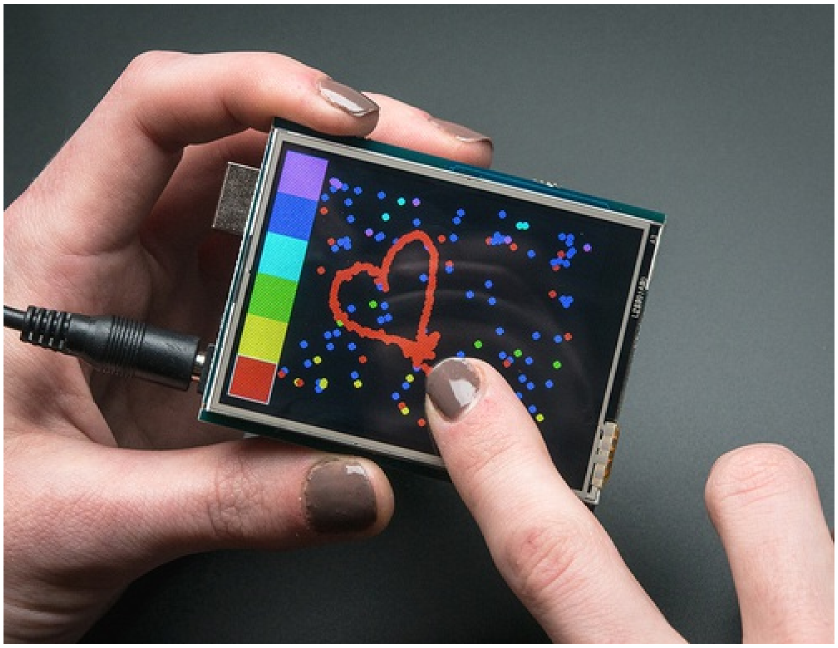
adafruit learning system

Adafruit 2.8" TFT Touch Shield v2

Created by Ladyada



Last updated on 2014-01-03 12:45:27 PM EST

Guide Contents

Guide Contents 2

Overview 3

Connecting 5

Using with an Uno 5

Using with a Mega/Leonardo 6

Graphics Test 8

Adafruit GFX library 10

Touchscreen Paint Demo 11

Drawing Bitmaps 14

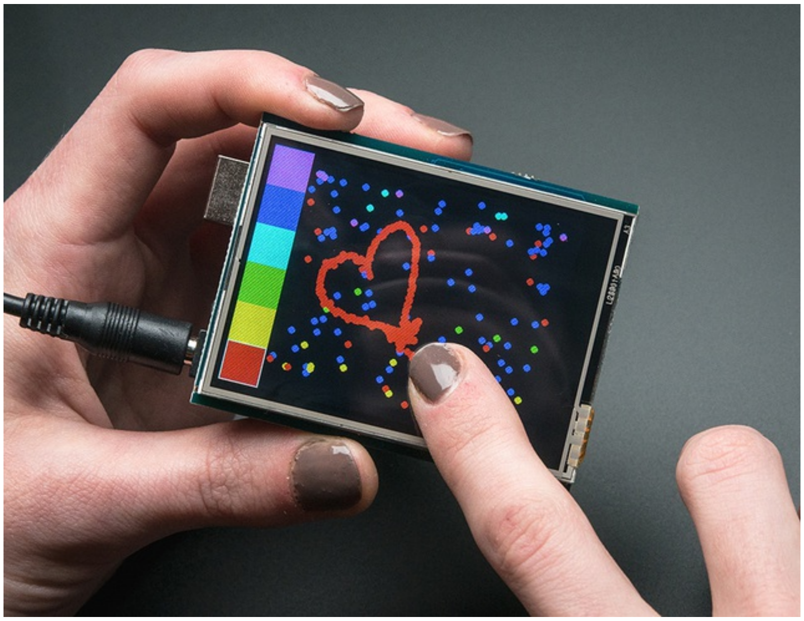
Backlight & Touch IRQ 16

Controlling the Backlight 16

Touchscreen Interrupt pin 16

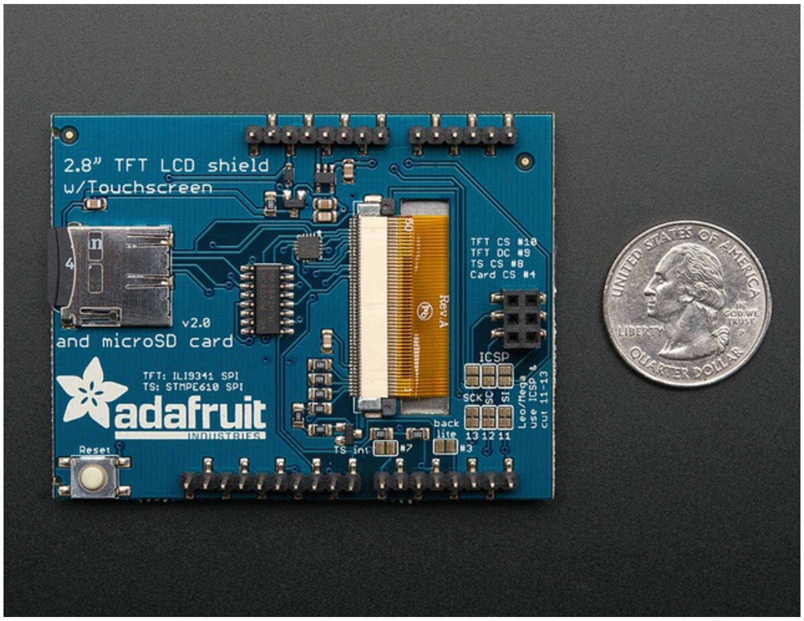
Downloads 17

Overview



Spice up your Arduino project with a beautiful large touchscreen display shield with built in microSD card connection. This TFT display is big (2.8" diagonal) bright (4 white-LED backlight) and colorful (18-bit 262,000 different shades)! 240x320 pixels with individual pixel control. It has way more resolution than a black and white 128x64 display. As a bonus, this display has a resistive touchscreen attached to it already, so you can detect finger presses anywhere on the screen.

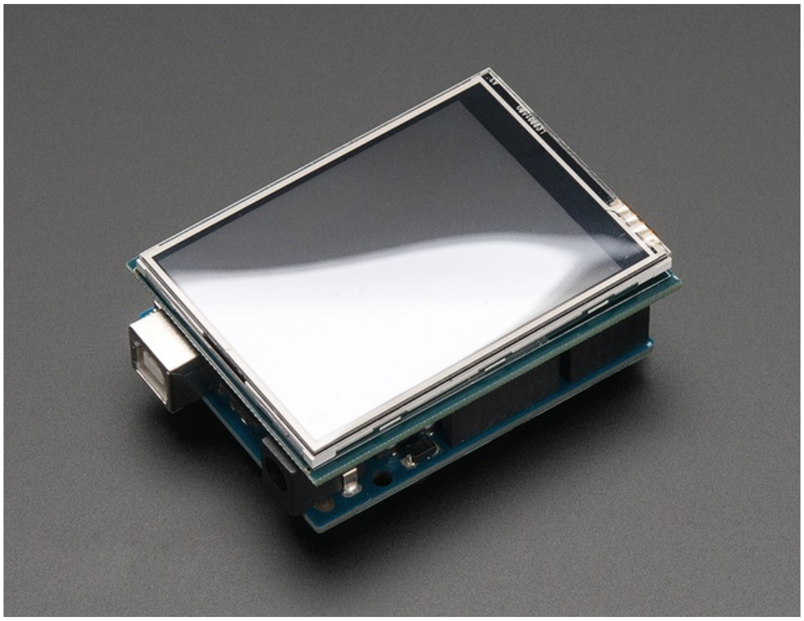
We've updated our original v1 shield to an SPI display - its a bit slower but uses a lot less pins and is now much easier to use with Mega & Leonardo. We also include an SPI touchscreen controller so you only need one additional pin to add a high quality touchscreen controller. Even with all the extras, the price is lower thanks to our parts sourcing & engineering skillz!



The shield is fully assembled, tested and ready to go. No wiring, no soldering! Simply plug it in and load up our library - you'll have it running in under 10 minutes! Works best with any classic Arduino (UNO/Duemilanove/Diecimila). Solder three jumpers and you can use it at full speed on a Leonardo or Mega as well.

This display shield has a controller built into it with RAM buffering, so that almost no work is done by the microcontroller. This shield needs fewer pins than our v1 shield, so you can connect more sensors, buttons and LEDs: 5 SPI pins for the display, another pin for the SPI touchscreen controller and another pin for uSD card if you want to read images off of it.

Connecting

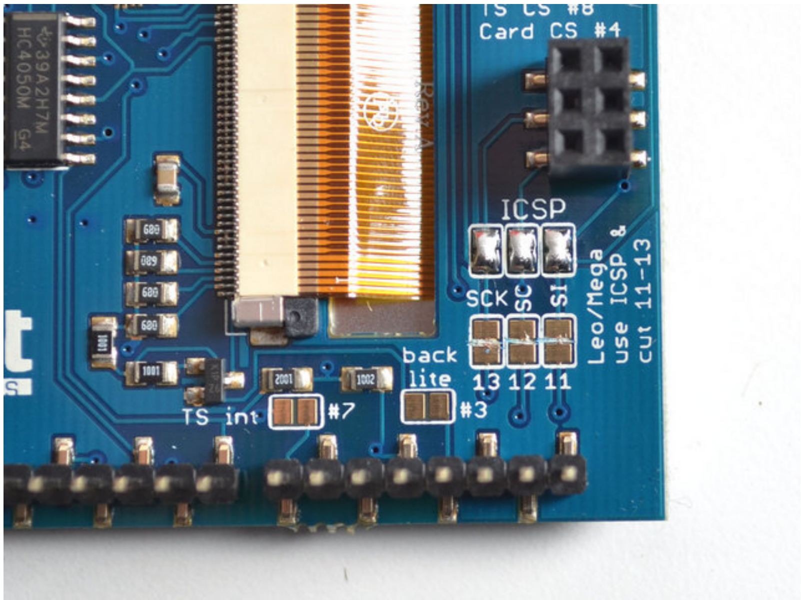


Using with an Uno

Because the TFT is about the same size as an Arduino, we pre-assemble the shield in the factory. To use, simply place it onto your Arduino Uno/Duemilanove/compatible. No wiring, no soldering! **Bam!**

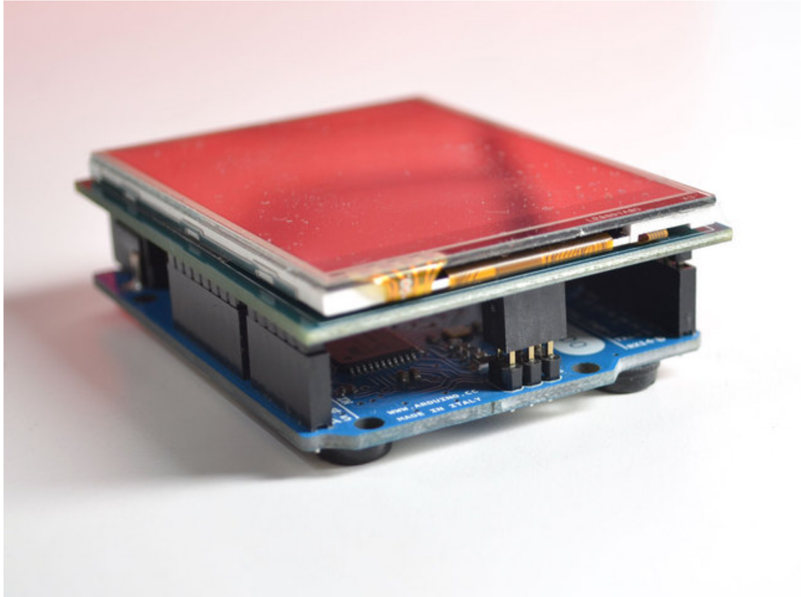
Using with a Mega/Leonardo

With just a little effort you can make this shield plug-n-play with the Mega/Leonardo. Look at the bottom of the shield and find the area with the 6 jumpers. Three of them are closed, and three are open. Solder close the three ICSP jumpers.



You can then cut the small line in between the #11, #12, #13 pads. This will free up digital #11, 12, and 13 for you to use on the Leo or Mega.

That's it! No software or firmware hacking other than that is required.



With this mod, the shield uses the ICSP header for data so make sure your duino (or stack of shields) has a ICSP plugged into the socket as above!

Graphics Test

We have a library with example code ready to go for use with these TFTs. The library is not incredibly fast and optimized but its a good start and can easily be ported to other micrcontrollers. However, we'll assume you're using an Arduino.

Our github repository (http://adafru.it/d4d) contains all the code and examples you'll need for driving the TFT. You can download the latest ZIP by clicking the button below

Download the Adafruit ILI9341 Library

http://adafru.it/d4e

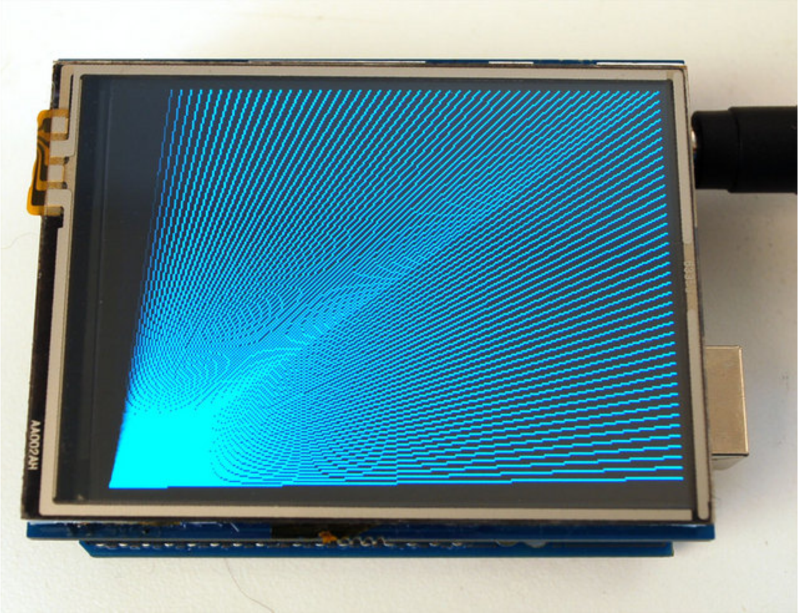
Uncompress the folder and rename it **Adafruit_ILI9341** make sure that inside that folder is the **Adafruit_ILI9341.cpp** and **Adafruit_ILI9341.h** files. Then copy it to your arduinosketchfolder/libraries folder. For more details, especially for first-time library installers, check out our great tutorial at http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use (http://adafru.it/aYM)

You're not done yet! You will also need to get the Adafruit GFX graphics core (http://adafru.it/aJa) - this is the general purpose graphics drawing library (the ILI9341 library just handles the low level stuff). Click this button to grab it if you don't have it yet

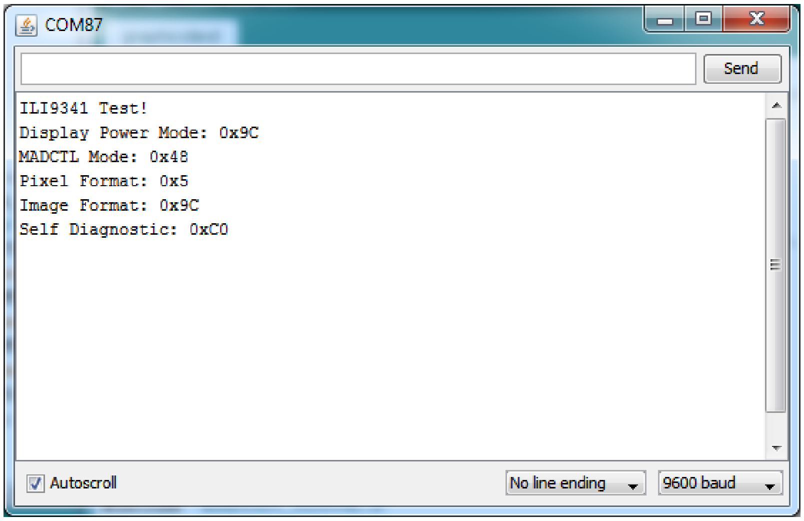
Download Adafruit GFX Library

http://adafru.it/cBB

Uncompress the folder and rename it **Adafruit_GFX** make sure that inside that folder is the cpp and .h files. Then copy it to your arduinosketchfolder/libraries folder like before



Restart the Arduino software. You should see a new **example** folder called **Adafruit_ILI9341** and inside, an example called **graphicstest**. Upload that sketch to your Arduino! You should see a collection of graphical tests draw out on the TFT.



```
ILI9341 Test!
Display Power Mode: 0x9C
MADCTL Mode: 0x48
Pixel Format: 0x5
Image Format: 0x9C
Self Diagnostic: 0xC0
```

☑ Autoscroll

No line ending

9600 baud

Adafruit GFX library

The TFT LCD library is based off of the Adafruit GFX graphics core library. GFX has many ready to go functions that should help you start out with your project. Its not exhaustive and we'll try to update it if we find a really useful function. Right now it supports pixels, lines, rectangles, circles, round-rects, triangles and printing text as well as rotation.

Check out the GFX tutorial for detailed information about what is supported and how to use it (http://adafru.it/aPx)!

Touchscreen Paint Demo

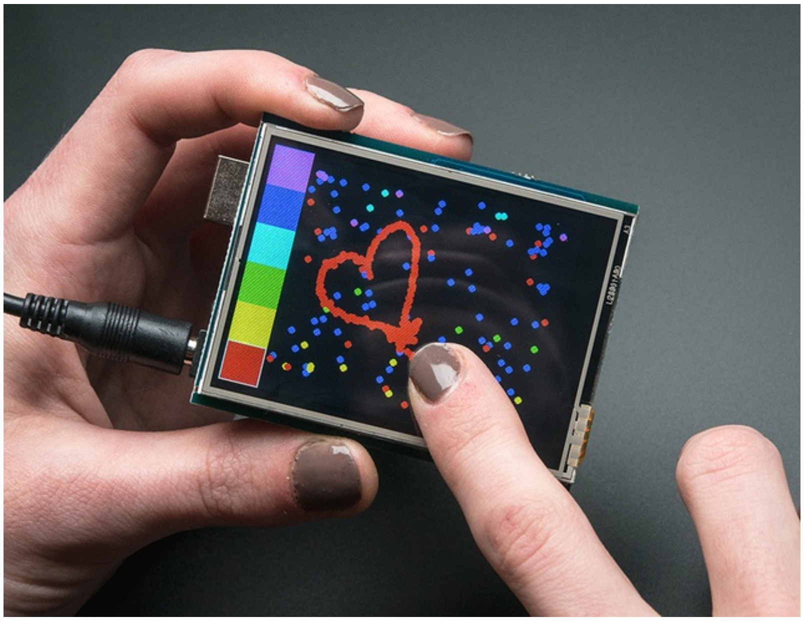
The LCD has a 2.8" 4-wire resistive touch screen glued onto it. You can use this for detecting finger-presses, stylus', etc. Normally, you'll need 4 pins to talk to the touch panel **but** we decided to go all snazzy and put a dedicated touch screen driver onto the shield. The driver shares the SPI pins with the TFT and SD card, so only one extra pin is needed (digital #8) This allows you to query the controller when you're ready to read touchscreen data, and saves 3 pins.

To control the touchscreen you'll need one more library (http://adafru.it/d4f) - the STMPE610 controller library which does all the low level chatting with the STMPE610 driver chip. Click below to download and then install it as before.

Download the STMPE610 Library

http://adafru.it/d4g

Once you have the library installed, restart the IDE. Now from the **examples->Adafruit_ILI9341** menu select **touchpaint** and upload it to your Arduino.



The touch screen is made of a thin glass sheet, and its very fragile - a small crack or break will make the entire touch screen unusable. Don't drop or roughly handle the TFT and be especially careful of the corners and edges. When pressing on the touchscreen, sometimes people can use the tip of their fingers, or a fingernail. If you don't find the touchscreen responds well to your fingers, you can use a rounded stylus which will certainly work. Do not press harder and harder until the screen cracks!

Getting data from the touchscreen is fairly straight forward. Start by creating the touchscreen object with

```
Adafruit_STMPE610 ts = Adafruit_STMPE610(STMPE_CS);
```

We're using hardware SPI so the clock, mosi and miso pins are not defined here. For the shield, CS is #8 always.
Then you can start the touchscreen with

```
ts.begin()
```

Check to make sure this returns a True value, which means the driver was found. If it wasn't, make sure you have the hardware SPI jumpers set up right: for Leonardo/Mega the ICSP jumpers get closed.

Now you can call

```
if (! ts.bufferEmpty()))
```

to check if there's any data in the buffer. The touchscreen driver will store touchpoints at all times. When you're ready to get the data, just check if there's any data in the buffer. If there is, you can call

```
TS_Point p = ts.getPoint();
```

To get the oldest point from the buffer. TS_Point has **.x .y** and **.z** data points. The **x** and **y** points range from 0 to 4095. The STMPE610 does not store any calibration data in it and it doesn't know about rotation. **So if you want to rotate the screen you'll need to manually rotate the x/y points!** The z point is 'pressure' and ranges from 0 to 255, we don't use it here but you can experiment with it on your own, the harder you press, the lower the number.

Since data from the STMPE610 comes in 0-4095 but our screen is 320 pixels by 240 pixels, we can use **map** to convert 0-4095 to 0-320 or 0-240. Something like

```
p.x = map(p.x, 0, 4095, 0, tft.width());
p.y = map(p.y, 0, 4095, 0, tft.height());
```

However, the touchscreen is a bit bigger than the screen, so we actually need to ignore presses beyond the touchscreen itself. We found that these numbers reflected the true range that overlaps the screen

```
#define TS_MINX 150
#define TS_MINY 130
#define TS_MAXX 3800
#define TS_MAXY 4000
```

So we use

```
p.x = map(p.x, TS_MINX, TS_MAXX, 0, tft.width());
p.y = map(p.y, TS_MINY, TS_MAXY, 0, tft.height());
```

instead.

One last point (pun intended!) since the touchscreen driver stores points in a buffer, you may want to ask the driver "is the touchscreen being pressed RIGHT NOW?" You can do that with

```
if (ts.touched())
```

Drawing Bitmaps

There is a built in microSD card slot into the shield, and we can use that to load bitmap images! You will need a microSD card formatted **FAT16 or FAT32** (they almost always are by default).

Its really easy to draw bitmaps. Lets start by downloading this image of pretty flowers (pix by johngineer)

Copy **purple.bmp** into the base directory of a microSD card and insert it into the microSD socket in the shield. Now upload the **file->examples->Adafruit_ILI9341->spitftbitmap** example to your Arduino + shield. You will see the flowers appear!

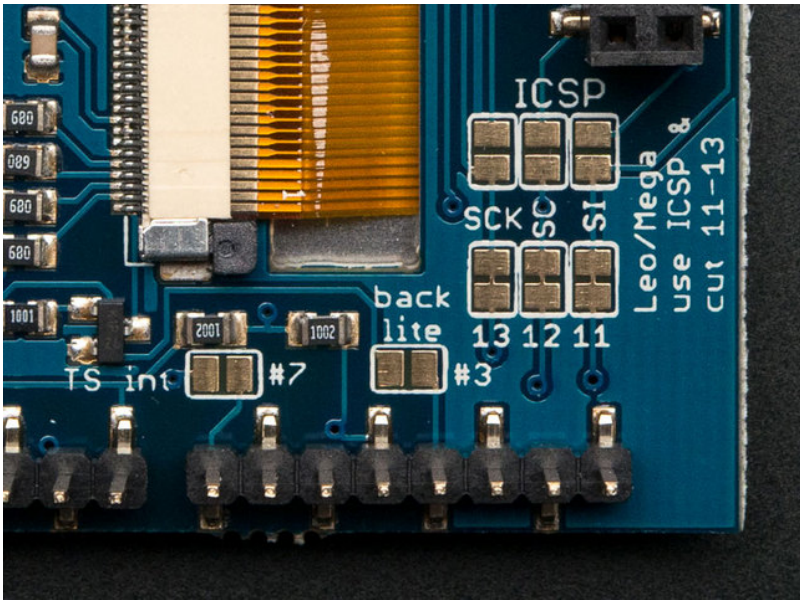
To make new bitmaps, make sure they are less than 240 by 320 pixels and save them in **24-bit BMP format**! They must be in 24-bit format, even if they are not 24-bit color as that is the easiest format for the Arduino. You can rotate images using the **setRotation()** procedure

You can draw as many images as you want - dont forget the names must be less than 8 characters long. Just copy the BMP drawing routines below loop() and call

```
bmpDraw(bmpfilename, x, y);
```

For each bitmap. They can be smaller than 320x240 and placed in any location on the screen.

Backlight & Touch IRQ



Controlling the Backlight

By default, we assume you'll want the backlight on all the time. However, you may want to PWM control or otherwise turn off the LED backlight to save power. You can do this with a simple hack. On the back, look for the **backlight** jumper. Solder the jumper labeled **Pin 3**. Then you can use Digital 3 to control the backlight.

Touchscreen Interrupt pin

Advanced users may want to get an interrupt on a pin (or even, just test a pin rather than do a full SPI query) when the touchscreen is pressed. You can do that by jumpering the #7 solder jumper labeled **TS int**. We didn't want it to connect to #2 or #3 since those are the Leonardo I2C pins. You can use pin change interrupts to get an interrupt callback on #7. (http://adafru.it/d4h)Or, with a little blue wire, advanced users can connect a wire from the TS interrupt pad to any pin they choose. We find that querying the

Downloads

Datasheets:

- STMPE610 (http://adafru.it/d4k)
- ILI9341 (TFT controller) (http://adafru.it/d4l)
- Raw 2.8" TFT datasheet (http://adafru.it/d4m)

Schematic of the v2 touchshield

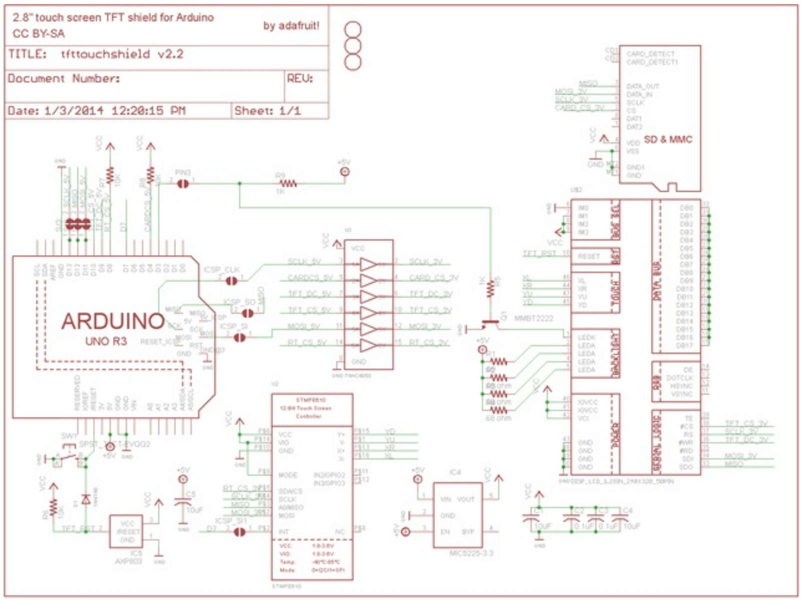
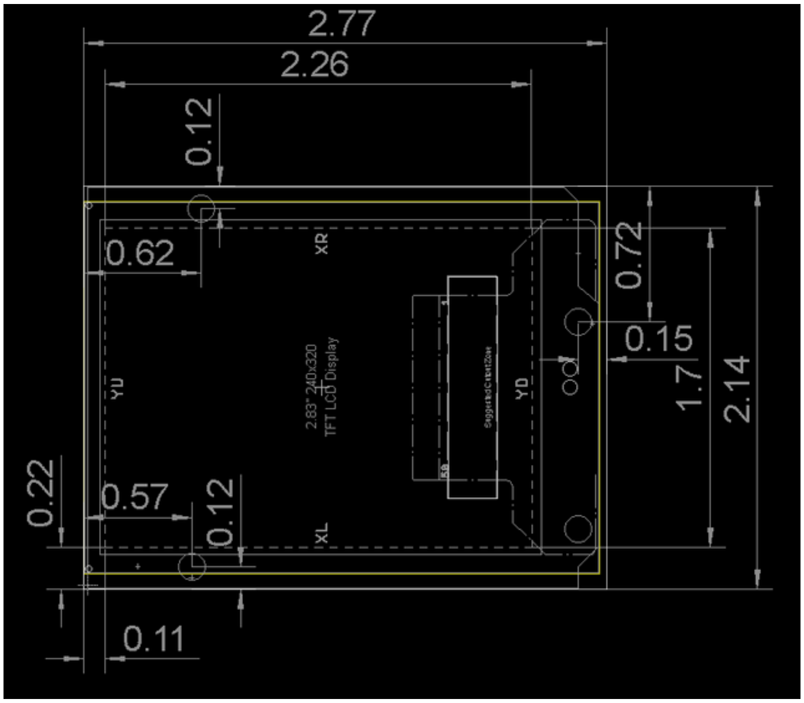


Diagram showing the TFT (yellow outline) underlying Arduino mounting holes (thin white line), PCP outline (rectangular thin white line) and 'visible portion' of the TFT (dashed inner line)


2.77
2.26
0.12
0.62
XR
0.72
0.15
1.7
2.14
YU
2.83" 240x320
TFT LCD Display
1
SuggestedCutoutZone
50
YD
0.22
0.57
0.12
XL
0.11