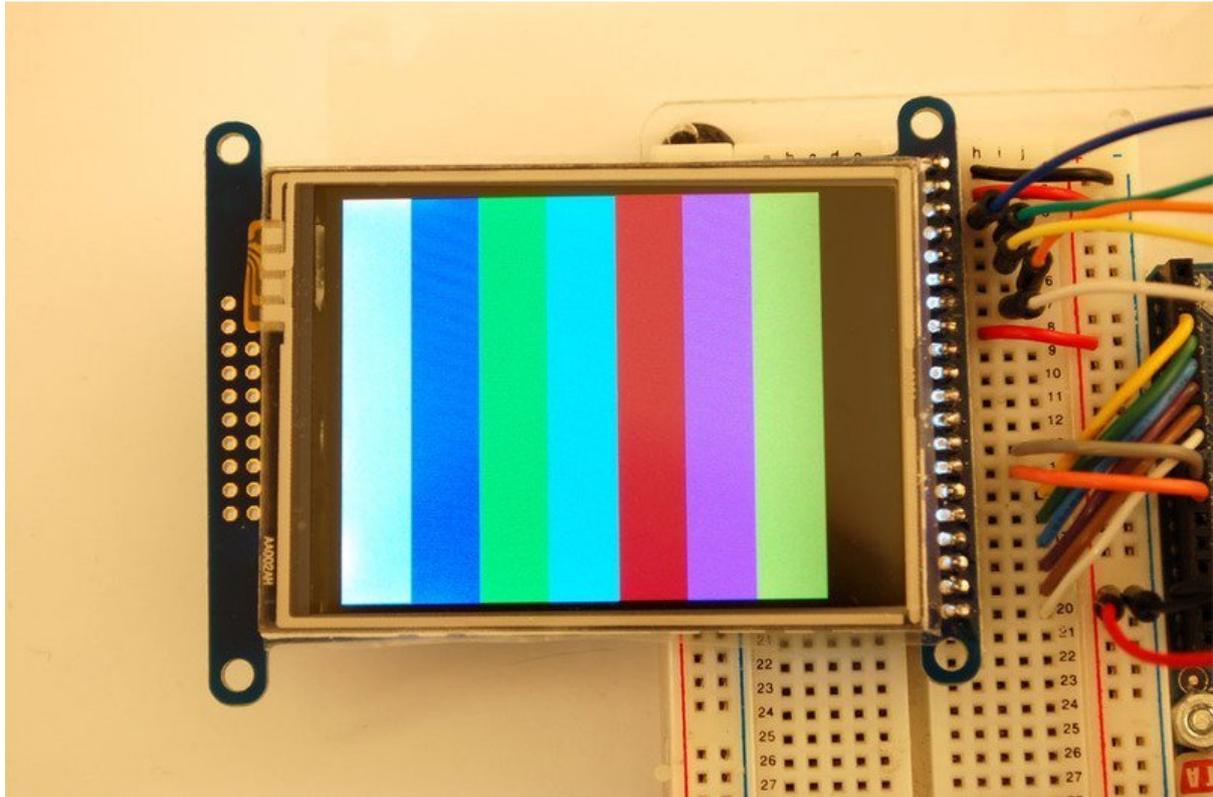




2.8" TFT Touchscreen

Created by lady ada



<https://learn.adafruit.com/2-8-tft-touchscreen>

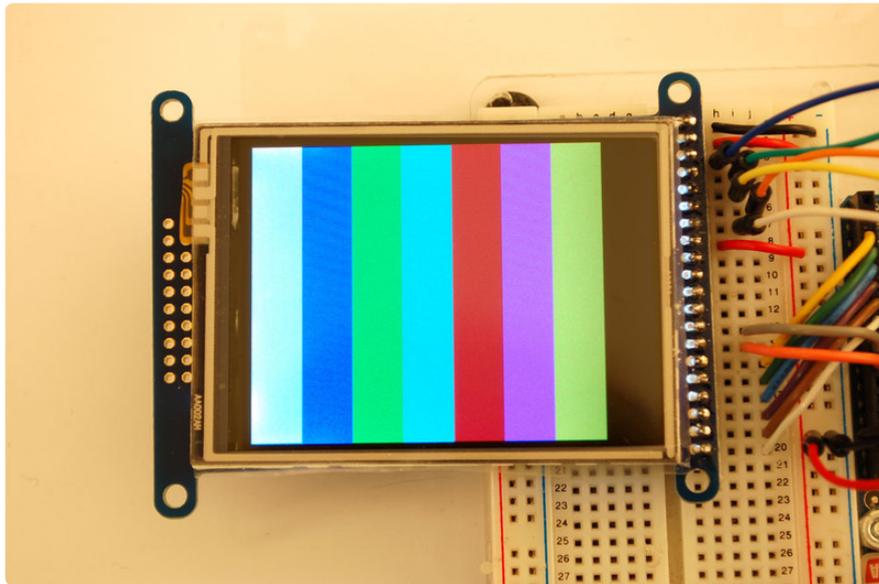
Last updated on 2024-06-03 01:13:49 PM EDT

Table of Contents

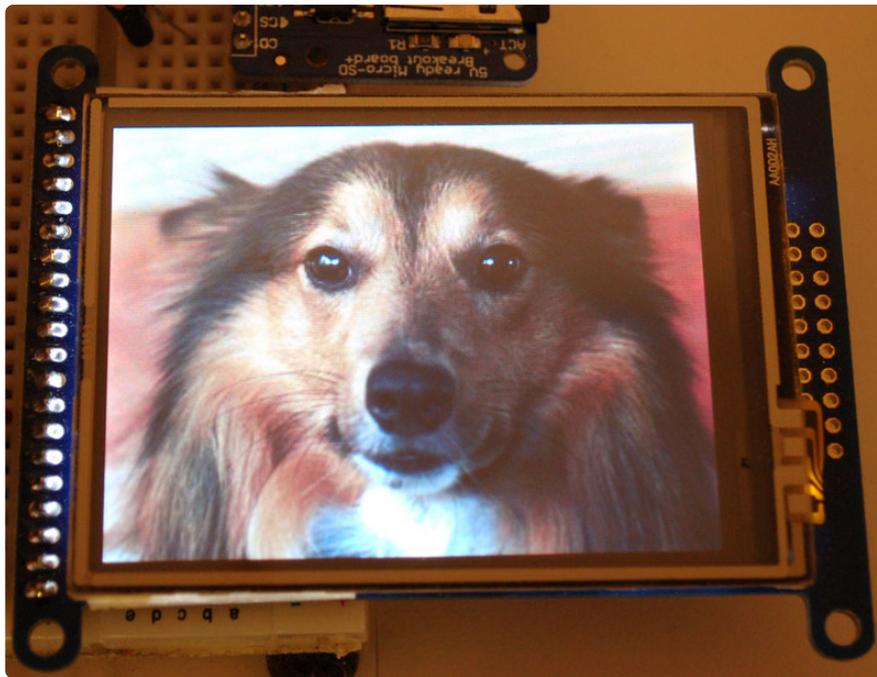
Overview	3
Connection Options	6
Mounting Options	8
Backlight Wiring	8
TFT Wiring	10
LCD Test	12
Graphics Library	15
Touchscreen	19
• Download Library	
Bitmaps	22
Downloads	23
Support Forums	24

Overview

This tutorial is for the original 2.8" TFT breakout, we have an updated tutorial now at <https://learn.adafruit.com/adafruit-2-dot-8-color-tft-touchscreen-breakout-v2> for the most recent version of this product, but are keeping this tutorial for historical reference



Add some jazz & pizzazz to your project with a color touchscreen LCD. This TFT display is big (2.8" diagonal) bright (4 white-LED backlight) and colorful (16-bit 262,000 different shades)! 240x320 pixels with individual pixel control, this 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.



This display has a controller built into it with RAM buffering, so that almost no work is done by the microcontroller. You'll need 8 digital data lines and 4 or 5 digital control lines to read and write to the display (12 lines total). 4 pins are required for the touch screen (2 digital, 2 analog) but because of the way resistive touch screens work, we can share pins with the LCD so the entire setup can be run by 12 pins (10 digital, 2 analog).

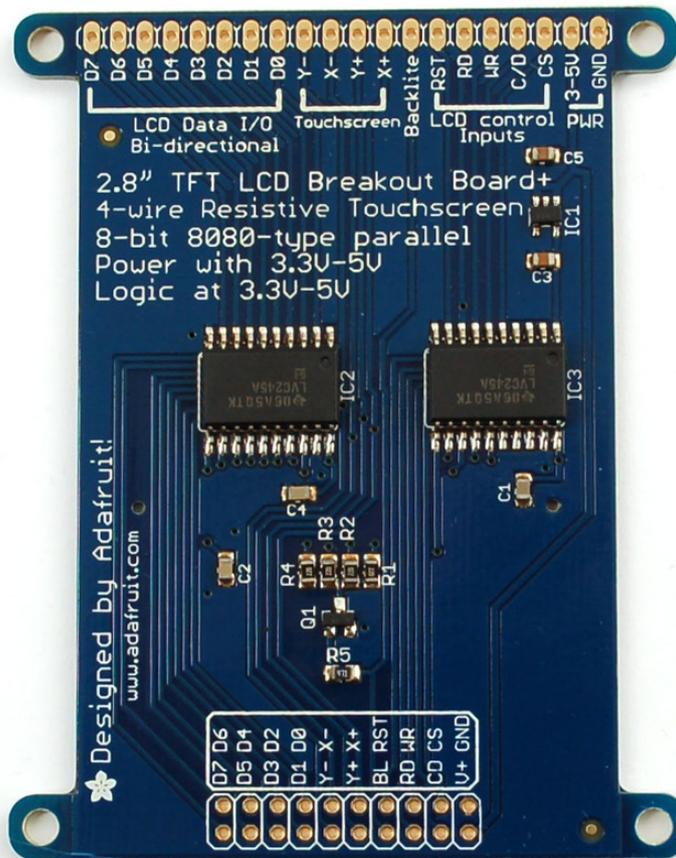
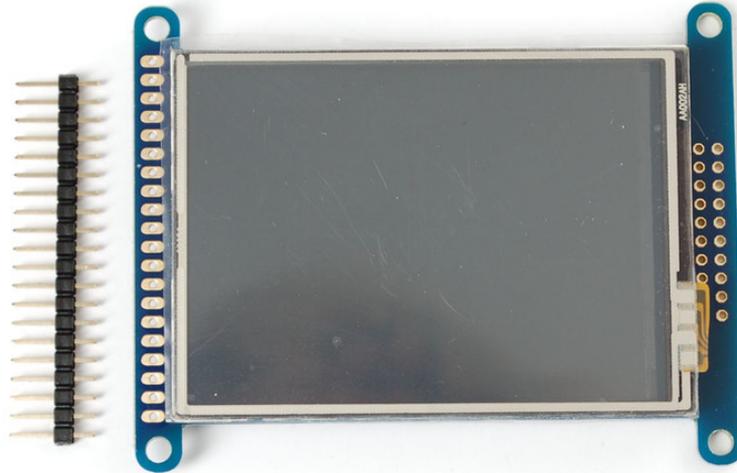
Of course, we wouldn't just leave you with a datasheet and a "good luck!" - [we've written a full open source graphics library that can draw pixels, lines, rectangles, circles and text](https://adafru.it/aHk) (<https://adafru.it/aHk>) . We [also have a touch screen library that detects x, y and z \(pressure\)](https://adafru.it/aT1) (<https://adafru.it/aT1>) and example code to demonstrate all of it. The code is written for Arduino but can be easily ported to your favorite microcontroller!

[Pick one up today at the Adafruit Shop! \(https://adafru.it/aIH\)](https://adafru.it/aIH)

Specifications:

- 2.8" diagonal LCD TFT display
- 240x320 resolution, 16-bit (262,000) color
- [ILI9325 \(datasheet\) \(https://adafru.it/aT2\)](https://adafru.it/aT2) or [ILI9328 \(datasheet\) \(https://adafru.it/aT3\)](https://adafru.it/aT3) controller with built in video RAM buffer
- 8 bit digital interface, plus 4 or 5 control lines
- 5V compatible! Use with 3.3V or 5V logic
- Onboard 3.3V @ 150mA LDO regulator
- 4 white LED backlight, transistor connected so you can PWM dim the backlight
- 1x20 header for easy breadboarding, or 2x10 header for cable connection

- 4 x 0.125"/3mm mounting holes with tabs
- 4-wire resistive touchscreen

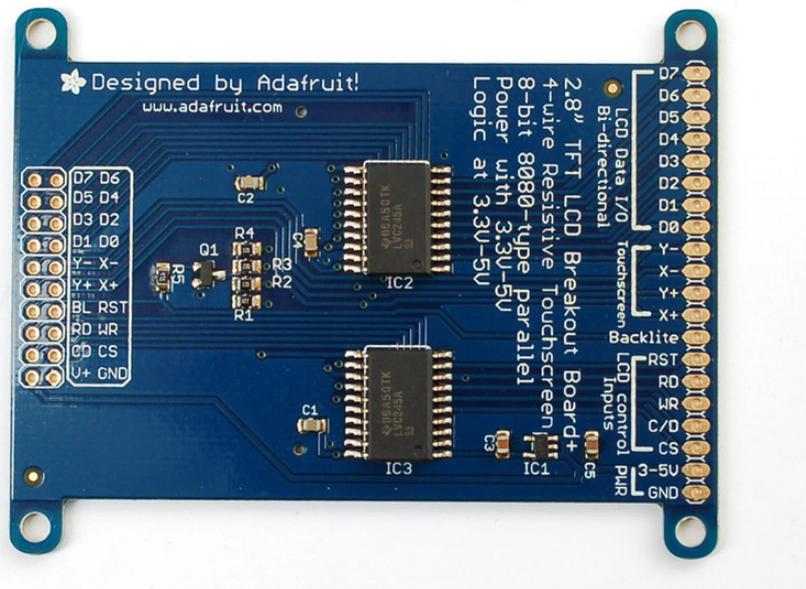


This guide is specifically for the TFT LCD breakout board. There's a [separate tutorial for the Arduino shield](https://adafru.it/aTQ) (<https://adafru.it/aTQ>) version of this display.

Connection Options

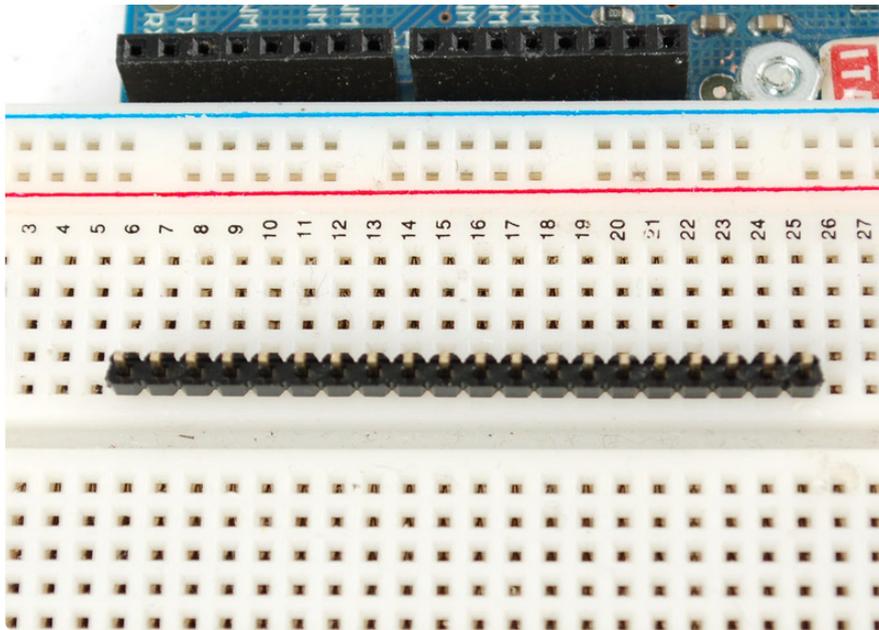
This tutorial is for the original 2.8" TFT breakout, we have an updated tutorial now at <https://learn.adafruit.com/adafruit-2-dot-8-color-tft-touchscreen-breakout-v2> for the most recent version of this product, but are keeping this tutorial for historical reference

The TFT LCD requires a lot of pins to use with your processor. We have two breakouts available which you can use depending on your application.

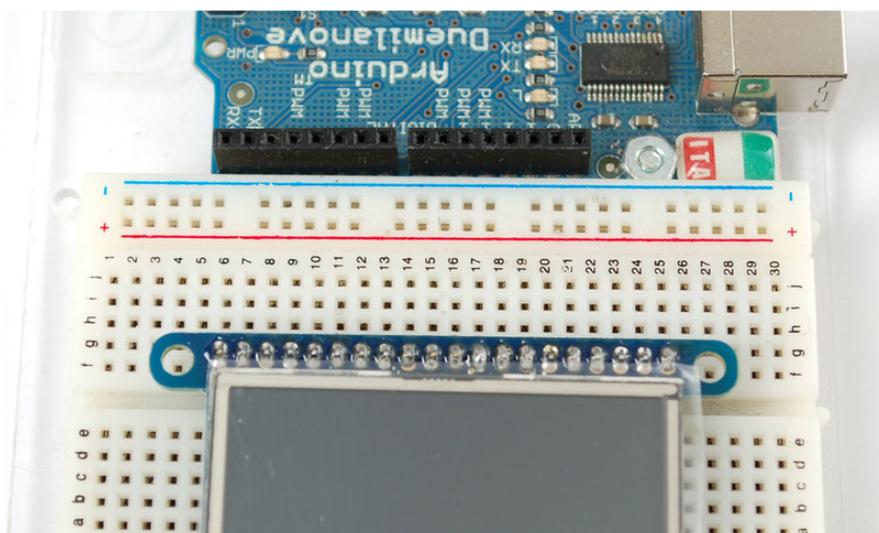
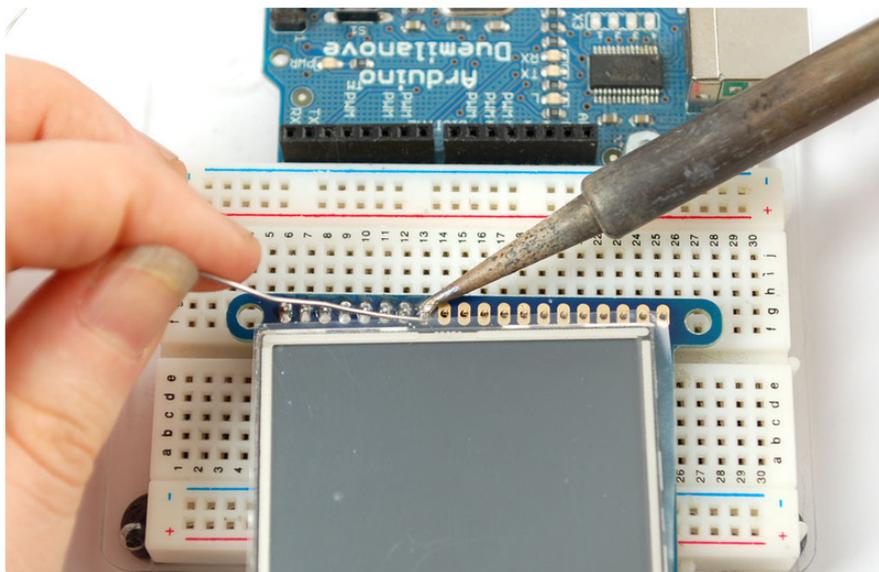


The first (right hand side) is a single 1x20 header strip with 0.1" spacing. This is perfect for breadboard use or if you have to do some hand wiring. You can use 'straight' header or 'right angle' header so that it stands up straight (although it may be harder to touch if its soldered this way).

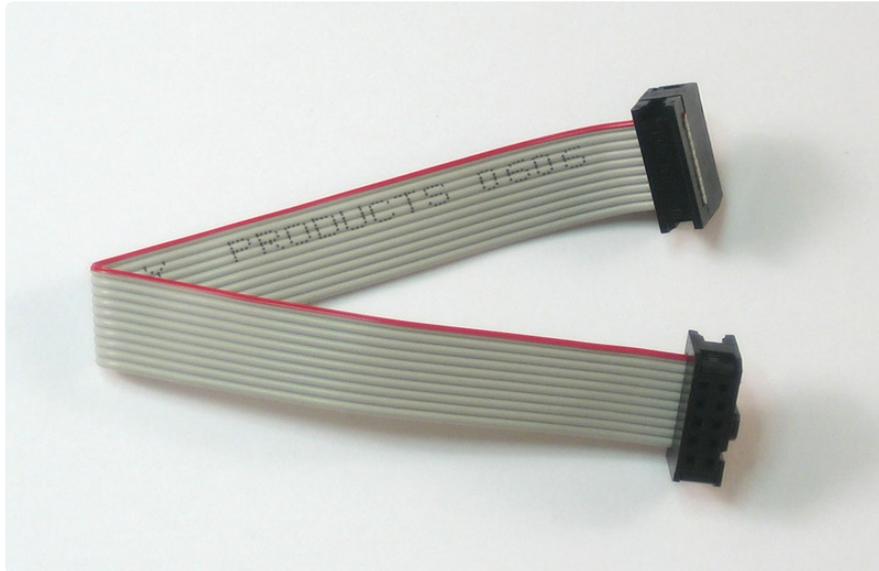
To solder the header, we suggest placing a 20pin strip of header, long pins down into a breadboard.



Place the LCD on top and solder each pin.



The second is a 2x10 dual header strip with 0.1" spacing. This is perfect for use with an IDC cable for remote access. We suggest a 6" cable but you might be able to use a 12" one, it depends on how fast you're writing to the screen (faster means a shorter cable required).



This cable is a 2x5 not 2x10 but its what the larger cable will look like

[You can pick up a socket-socket cable from digikey for \\$2 - they're also available from any other electronics shop \(https://adafru.it/aLu\).](https://adafru.it/aLu)

Mounting Options

This tutorial is for the original 2.8" TFT breakout, we have an updated tutorial now at <https://learn.adafruit.com/adafruit-2-dot-8-color-tft-touchscreen-breakout-v2> for the most recent version of this product, but are keeping this tutorial for historical reference

We wanted to make sure that you could easily put this display in a box. There are four mounting holes on tabs. If you really don't need the tabs they can be cut off with a hacksaw or tin-snips. The holes are 2.25" apart in the short direction and 2.95" apart in the long direction. The drill holes are 0.125" and will easily take a #4 imperial or M3 (3mm) screw. The PCB is 0.063" thick.

Backlight Wiring

This tutorial is for the original 2.8" TFT breakout, we have an updated tutorial now at <https://learn.adafruit.com/adafruit-2-dot-8-color-tft-touchscreen-breakout>

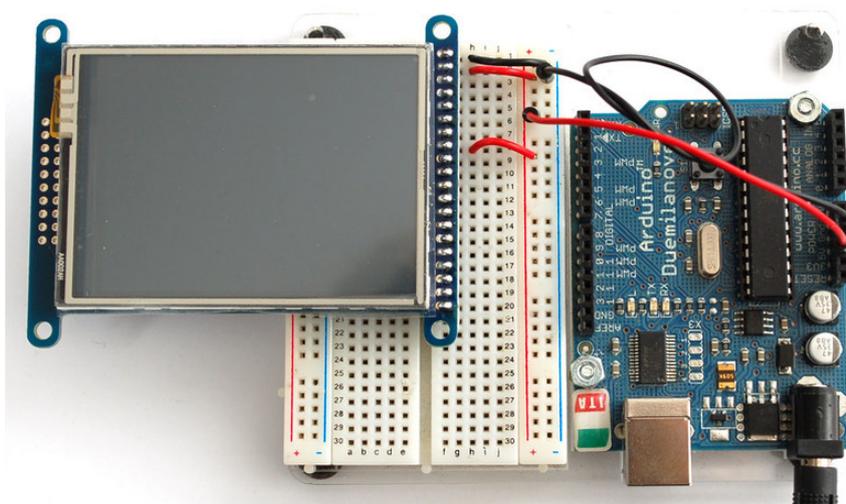
v2 for the most recent version of this product, but are keeping this tutorial for historical reference

There are three basic 'components' to the LCD.

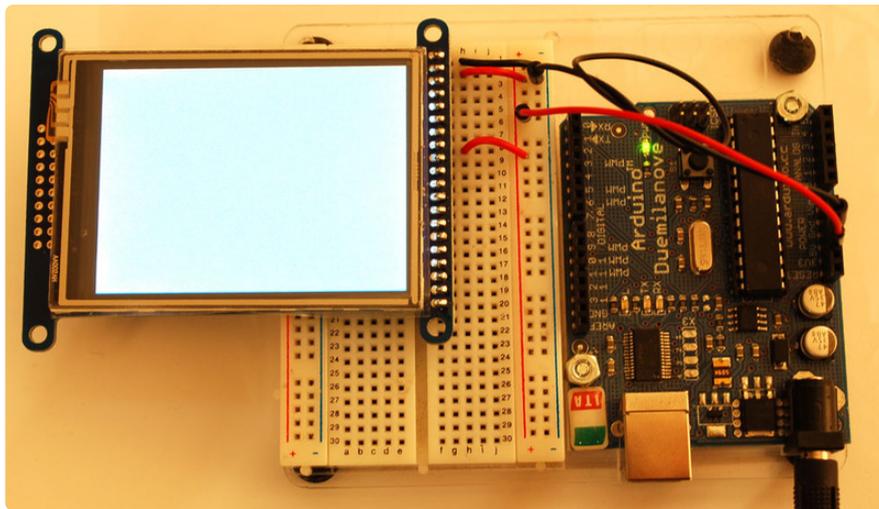
The backlight is the first. It is made of 4 white LEDs in parallel with a transistor to control them. The LEDs can draw as much as 80mA all together, but you can PWM the backlight to dim it - the transistor makes it easy to connect any kind of microcontroller output. You can also connect the backlight pin directly to a 3 or 5V pin to turn it on all the way.

We'll begin by assuming you'll be using the 1x20 connector and an Arduino. Wiring may be different for your microcontroller.

Start by connecting the first pin **Ground** of the LCD to ground and the second pin **3-5V** to 5V (you can use 3-5V, this pin will power the TFT and backlight so be sure it can supply 100mA). Then skip 5 pins and connect pin #8 **Backlite** to 5V.



Power up your set, and you'll see the 4 white LED backlight. If this isn't working, something is amiss with your power supply. Go back and fix the wiring!

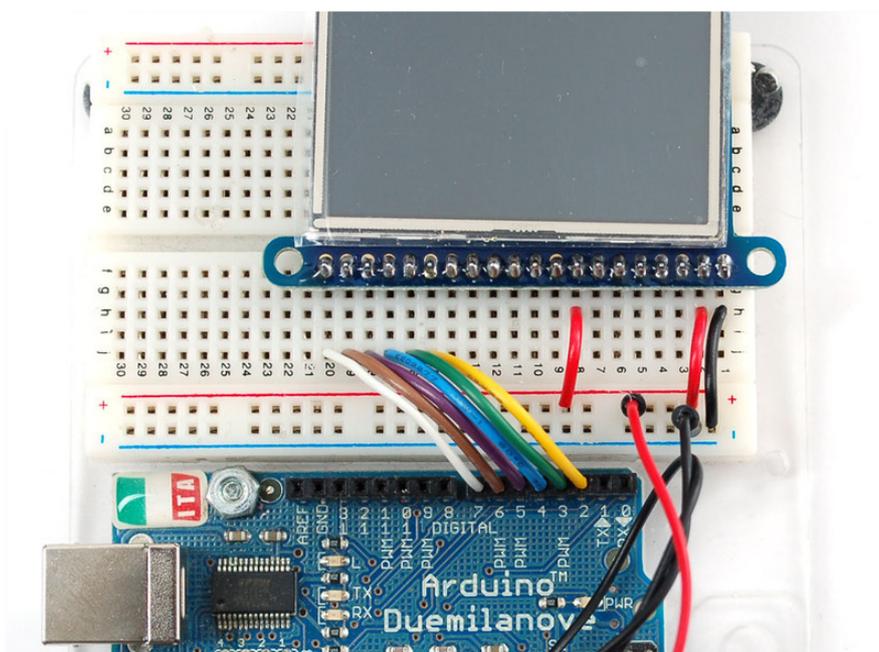


TFT Wiring

This tutorial is for the original 2.8" TFT breakout, we have an updated tutorial now at <https://learn.adafruit.com/adafruit-2-dot-8-color-tft-touchscreen-breakout-v2> for the most recent version of this product, but are keeping this tutorial for historical reference

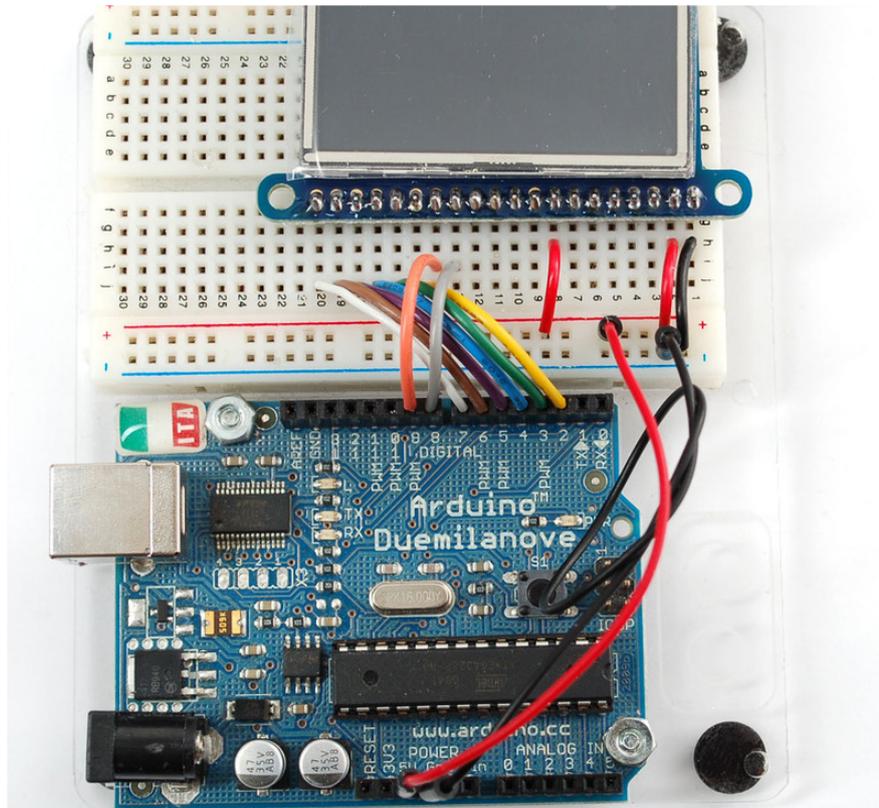
Now that the backlight is working, we can get the TFT LCD working. There are many pins required, and to keep the code running fairly fast, we have 'hardcoded' Arduino digital pins #2-#9 for the 8 data lines.

Start at the end of the TFT (other side than the power pins) and in order connect the pins to digital 7 thru 2. If you're using a mega, connect the TFT Data Pins #0-7 to Mega pins #22-29, in that order. Those Mega pins are on the 'double' header.



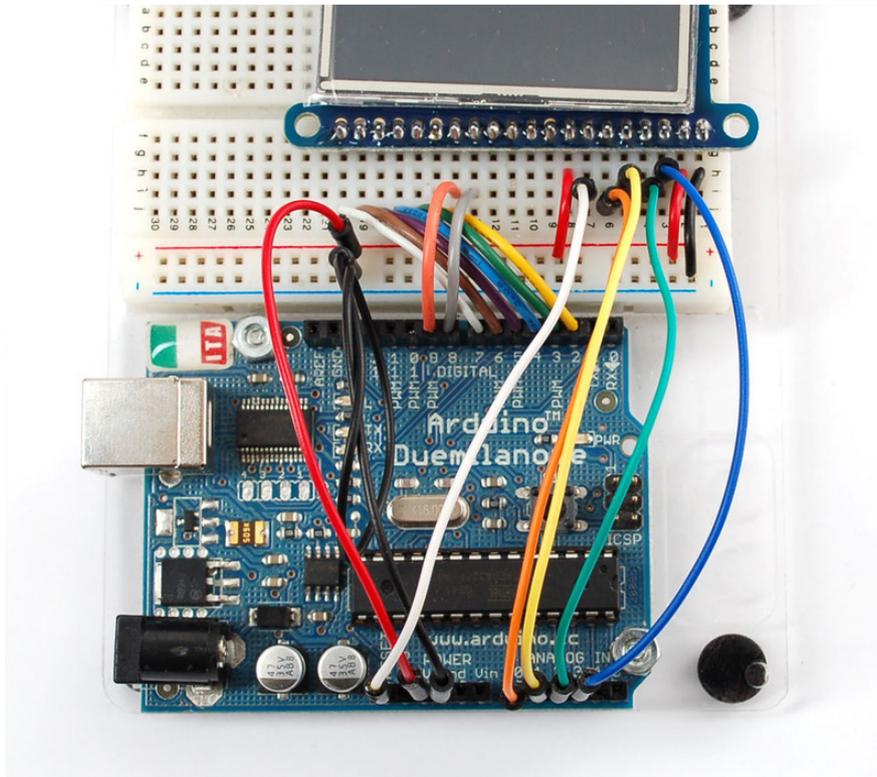
Then connect the next two pins to digital 9 and 8.

If you're using a mega, connect the TFT Data Pins #0-7 to Mega pins #22-29, in that order. Those Mega pins are on the 'double' header.



In addition to the 8 data lines, you'll also need 4 or 5 control lines.

- Connect the third pin **CS** (Chip Select) to Analog 3
- Connect the fourth pin **C/D** (Command/Data) to Analog 2
- Connect the fifth pin **WR** (Write) to Analog 1
- Connect the sixth pin **RD** (Read) to Analog 0
- Connect the seventh pin **RST** (Reset) to the Arduino Reset line. This will reset the panel when the Arduino is Reset. You can also use a digital pin for the LCD reset but this will save us a pin.



Now we can run some code!

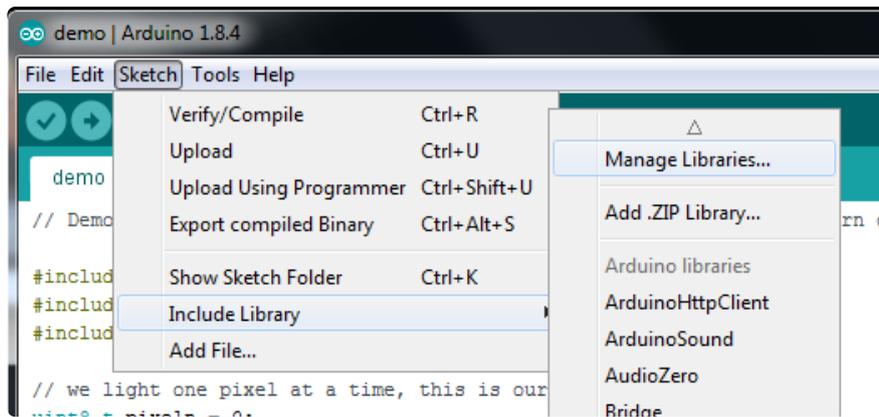
LCD Test

This tutorial is for the original 2.8" TFT breakout, we have an updated tutorial now at <https://learn.adafruit.com/adafruit-2-dot-8-color-tft-touchscreen-breakout-v2> for the most recent version of this product, but are keeping this tutorial for historical reference

We have example code ready to go for use with these TFTs. It's written for Arduino, which should be portable to any microcontroller by adapting the C++ source.

Two libraries need to be downloaded and installed: the [TFTLCD library \(https://adafru.it/aHk\)](https://adafru.it/aHk) and the [GFX library. \(https://adafru.it/aJa\)](https://adafru.it/aJa) You can install these libraries through the Arduino library manager.

Open up the Arduino library manager:



Search for the **Adafruit GFX** library and install it



Search for the **Adafruit TFTLCD** library and install it



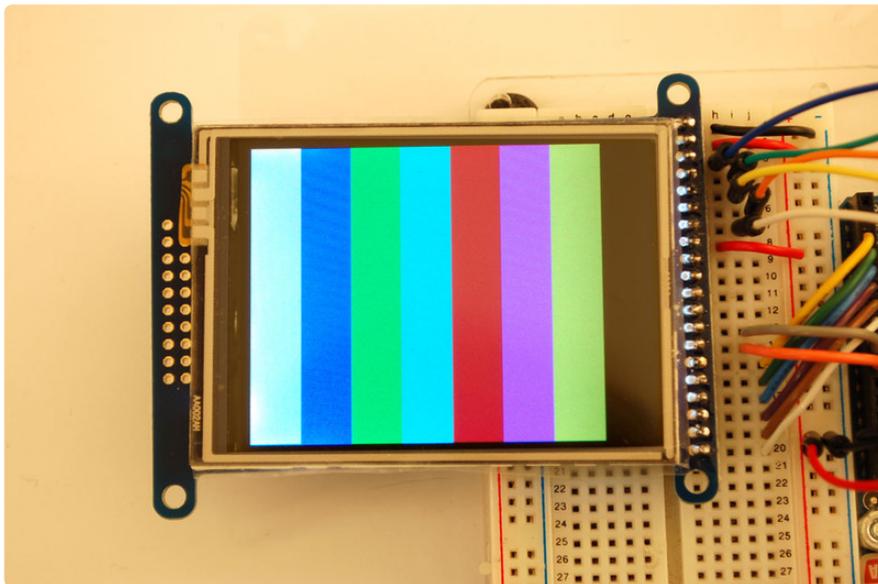
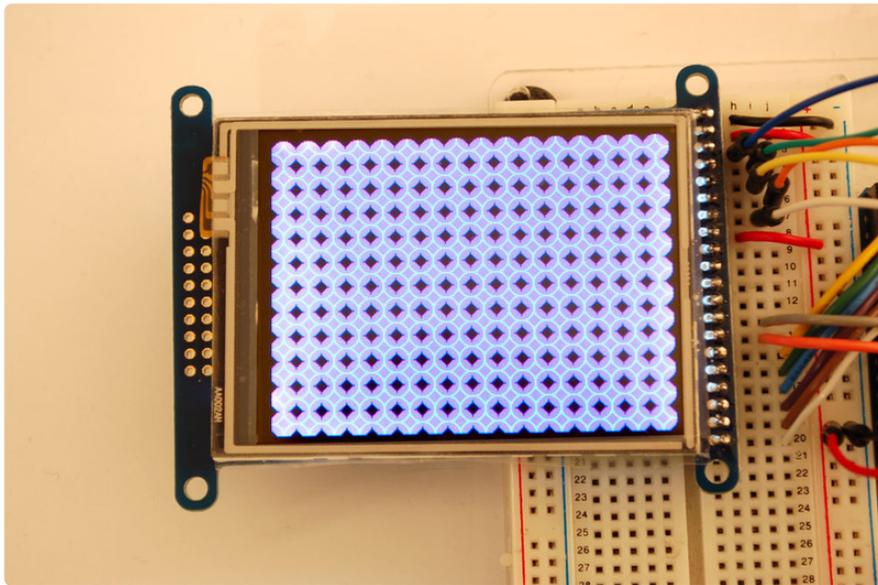
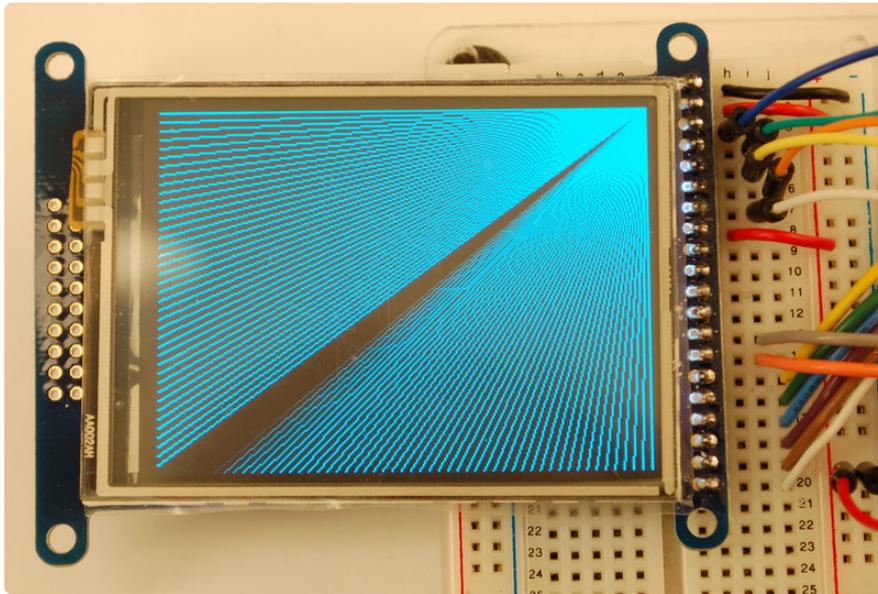
If using an older version of the Arduino IDE (pre-1.8.10), also locate and install the **Adafruit_BusIO** library (newer versions do this automatically when using the Arduino Library Manager).

We also have a great tutorial on Arduino library installation at:

<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use> (<https://adafru.it/aYM>)

In the TFTLCD Library folder, you will need to edit TFTLCD.h. On about line 12, you will see `"#define USE_ADAFRUIT_SHIELD_PINOUT"`. Comment out this line and save the file.

After restarting the Arduino software, you should see a new **example** folder called **Adafruit_TFTLCD** and inside, an example called **graphicstest**. Upload that sketch to your Arduino. You may need to press the Reset button to reset the arduino and TFT. You should see a collection of graphical tests draw out on the TFT.



Now you know everything is working. If you're having problems, check the serial port monitor. The first thing the sketch does is read the driver code from the TFT. It should be **0x9328** or **0x9325** so if you see something like **0x8328** that means that the D8 pin is not wired correctly and if you get **0x9228** then pin D0 is not wired correctly. Remember that the two bytes are read one after another.

Graphics Library

This tutorial is for the original 2.8" TFT breakout, we have an updated tutorial now at <https://learn.adafruit.com/adafruit-2-dot-8-color-tft-touchscreen-breakout-v2> for the most recent version of this product, but are keeping this tutorial for historical reference

The graphics library has a few 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.

First thing to note is that color is 16-bit, and that includes **Red**, **Green** and **Blue** in a 16-bit variable. The way the color is packed in is the top 5 bits are red, the middle 6 bits are green and the bottom 5 bits are blue.

For solid colors, we have this handy cheat-sheet. Of course, you can pick any of 262,000 colors but while starting out, this might be helpful.

```
// Color definitions
#define BLACK      0x0000
#define BLUE       0x001F
#define RED        0xF800
#define GREEN      0x07E0
#define CYAN       0x07FF
#define MAGENTA    0xF81F
#define YELLOW     0xFFE0

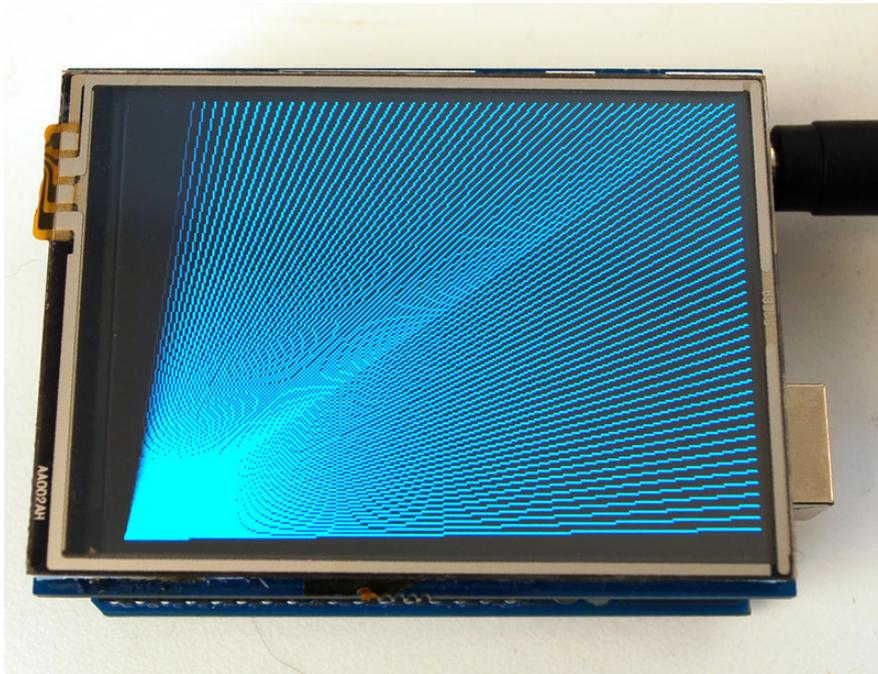
#define WHITE      0xFFFF
```

The Adafruit GFX library is what draws all the dots, lines, shapes, and text. Its fairly detailed and has lots of cool stuff, for lots of info, check out <http://learn.adafruit.com/adafruit-gfx-graphics-library> (<https://adafru.it/aPx>)

Here is a basic introduction to the GFX lib, it doesn't cover everything but it shows you what some of the most popular shapes look like

First up is the most basic pixel pusher. You can call this with two coordinates and a color and it will make a dot:

```
void drawPixel(uint16_t x, uint16_t y, uint16_t color);
```

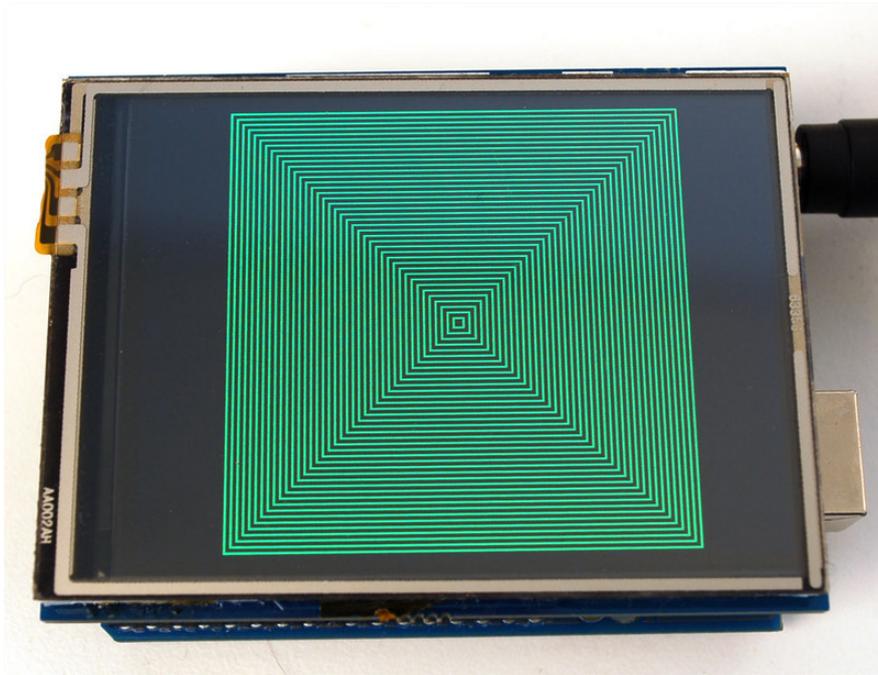


You can also draw lines, with a starting and end point and color.

```
void drawLine(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t color);
```

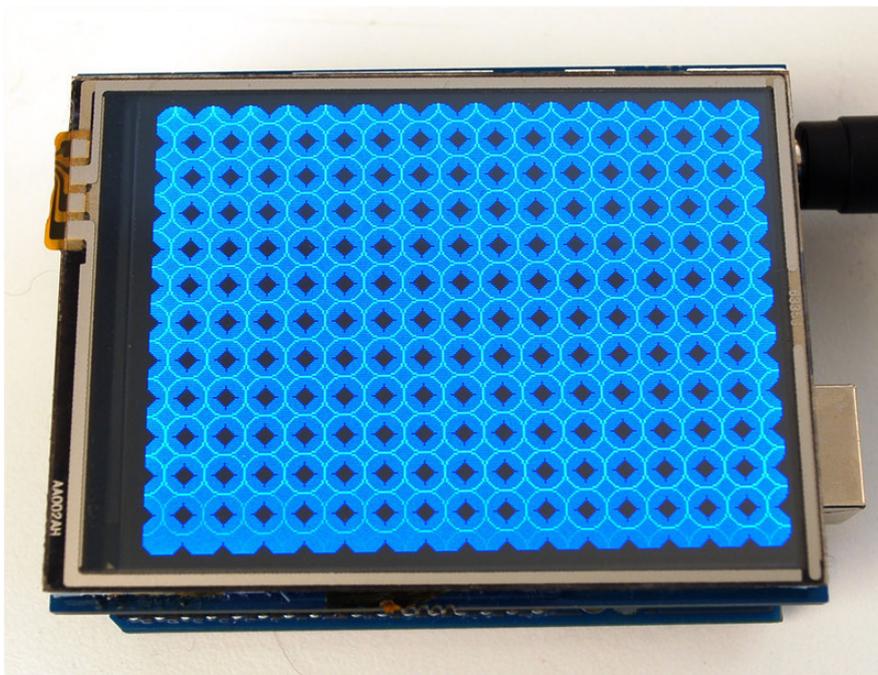
If your lines are vertical or horizontal, you can call an optimized drawing function that doesn't do all the angular calculations.

```
void drawVerticalLine(uint16_t x0, uint16_t y0, uint16_t length, uint16_t color);  
void drawHorizontalLine(uint16_t x0, uint16_t y0, uint16_t length, uint16_t color);
```



Next up, rectangles and squares can be drawn and filled using the following procedures. If you want a rectangle that has a contrasting outline color, **fillRect** first, then **drawRect** over it.

```
void drawRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t color);  
void fillRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t color);
```



Likewise, for circles, you can draw and fill.

```
void drawCircle(uint16_t x0, uint16_t y0, uint16_t r,    uint16_t color);  
void fillCircle(uint16_t x0, uint16_t y0, uint16_t r,    uint16_t color);
```



Text is a little different. Instead of one procedure, you will set up the text size, color and location and then **print()** (just like `Serial.print()`!)

```
void setCursor(uint16_t x0, uint16_t y0);  
void setTextColor(uint16_t color);  
  
void setTextSize(uint8_t size);
```

First start with **setCursor(x, y)** this will place the top right corner of the text wherever you please. Initially, its set to (0, 0). Then set the text color with **setTextColor(color)** by default its white. Then set the 'size' with **setTextSize(size)** this will 'multiply' the text by a scaling factor. Above you can see scales of 1 (default), 2 and 3. This is because we only ship the library with a simple font, to save space. You can just scale it to get bigger text without requiring a new font.

Finally, you can use **print()** or **println()** just like you do with **Serial**! For example, to print a string, use **print("Hello world")** - that's the first line of the image above. To print variables, you can also use **print()** the second line is **print(1234.56)** and the third line is **print(0xDEADBEEF, HEX)**.

You can also rotate your drawing. Note that this will not rotate what you already drew, but it will relocate any new drawing.



```
void rotate(uint8_t rotation);
```

The rotation variable can be 0, 1, 2 or 3. Rotation 0 makes it so that the display is in portrait mode, with the USB jack in the top right. Rotation 2 is portrait, with the USB jack in the bottom left. Rotation 1 is landscape mode, with the USB jack in the bottom right and rotation 3 is also landscape, with the USB jack in the top left.

When you rotate, the origin point moves with you. You may need to reference the size of the screen, which changes between portrait and landscape, use **width()** and **height()**! To get the size.

```
uint16_t width();  
uint16_t height();
```

These primitives should get you started!

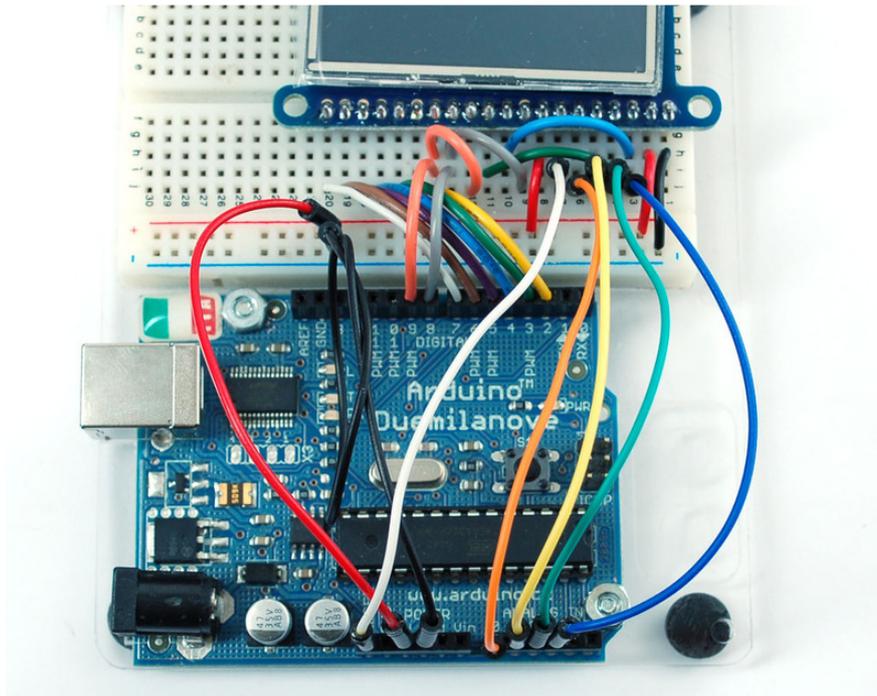
Touchscreen

This tutorial is for the original 2.8" TFT breakout, we have an updated tutorial now at <https://learn.adafruit.com/adafruit-2-dot-8-color-tft-touchscreen-breakout-v2> for the most recent version of this product, but are keeping this tutorial for historical reference

The LCD has a 2.8" 4-wire resistive touch screen glued onto it. You can use this for detecting finger-presses, stylus', etc. You'll need 4 pins to talk to the touch panel **but** you can **reuse** some of the pins for the TFT LCD! This is because the resistance of the

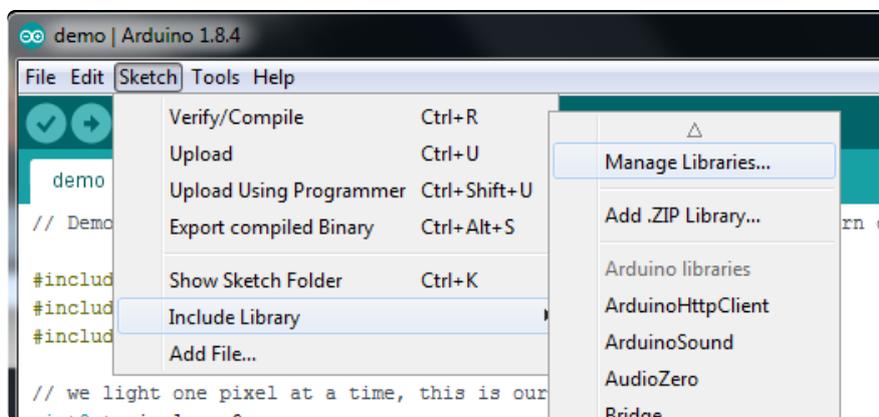
panel is high enough that it doesn't interfere with the digital input/output and we can query the panel in between TFT accesses, when the pins are not being used.

You can wire up the 4 remaining pins as follows. the one on the very left (Y- orange) can connect to digital **9**, the next one over (X- green) connects to **Analog 2**, The next one over (Y+ blue) connects to **Analog 3** and the last one (X+ gray) connects to digital **8**. The X- and Y+ pins pretty much have to connect to those analog pins (or to analog 4/5) but Y-/X+ can connect to any digital or analog pins.



Download Library

Begin by opening up the Arduino library manager



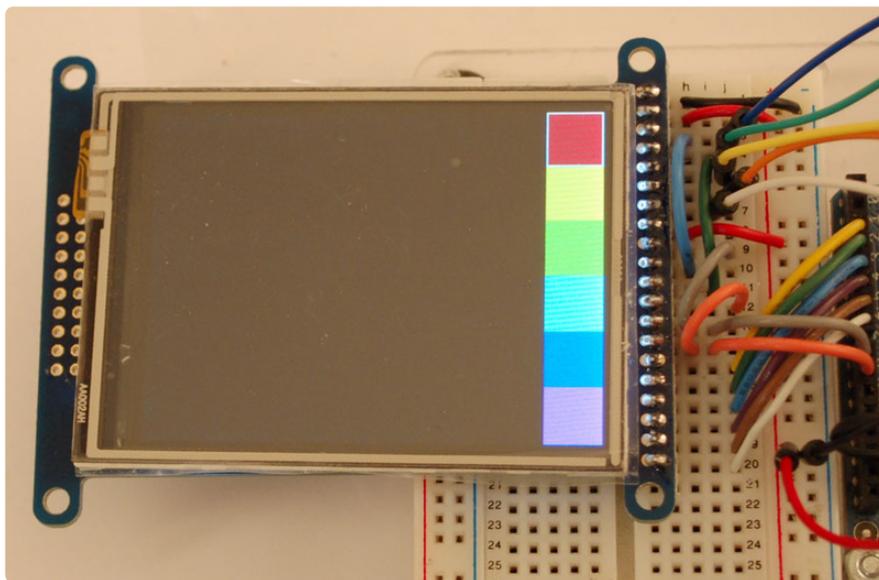
Search for the **Adafruit TouchScreen** library and install it



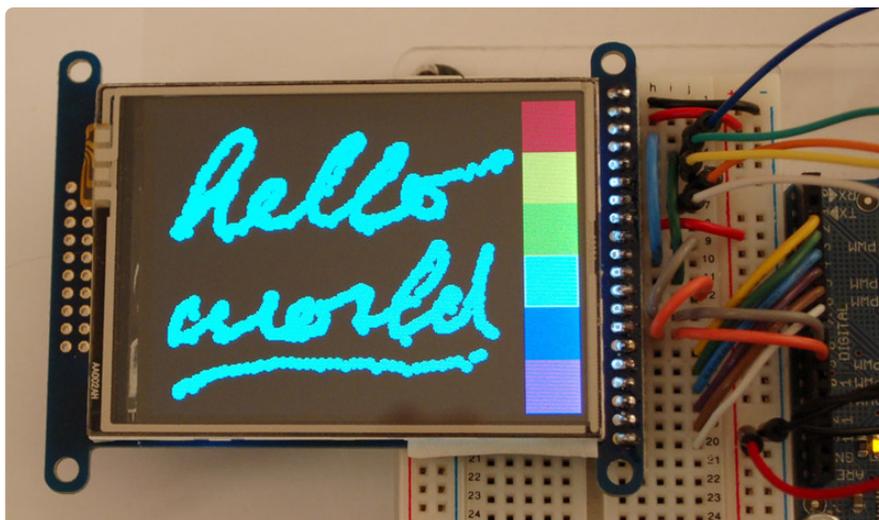
We also have a great tutorial on Arduino library installation at:

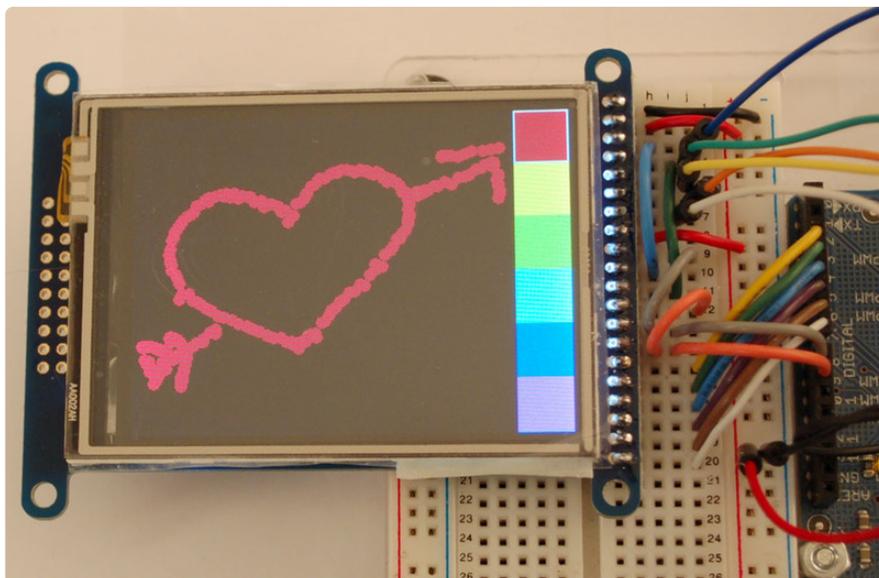
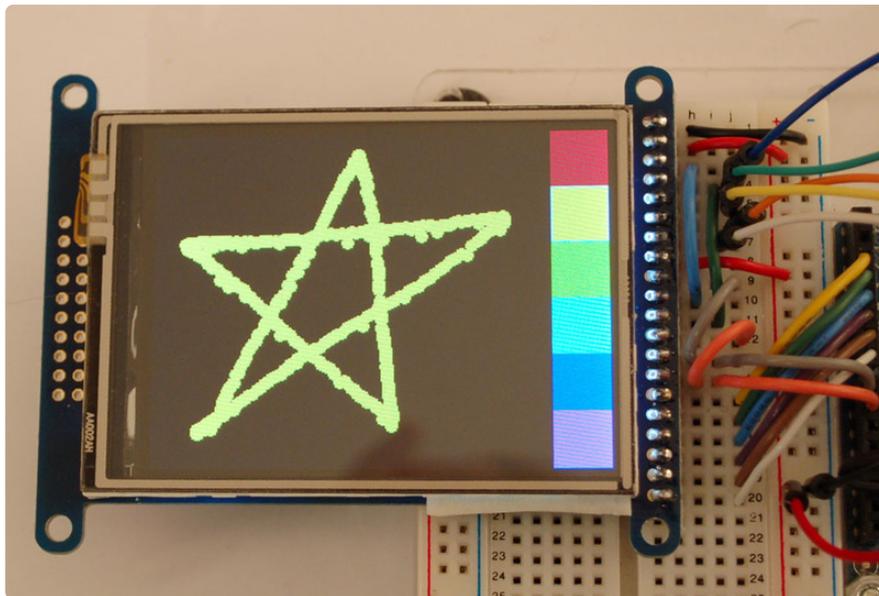
<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use> (<https://adafru.it/aYM>)

Now start up the **tftpaint** example in the Arduino library. The right hand side will have 'color boxes' you can press to select which color you want to draw with. If you press the area to the **left** where the screen ends, it will erase the screen.



You can use your fingertip to draw.





If you press the area to the **left** where the screen ends, it will erase the screen.

Bitmaps

This tutorial is for the original 2.8" TFT breakout, we have an updated tutorial now at <https://learn.adafruit.com/adafruit-2-dot-8-color-tft-touchscreen-breakout-v2> for the most recent version of this product, but are keeping this tutorial for historical reference

We have an example sketch in the library showing how to display full color bitmap images stored on an SD card. You'll need an SD or [microSD breakout board such as this one \(https://adafru.it/a1H\)](https://adafru.it/a1H).

Modern versions of the Arduino software (0023, or 1.0 or later) already include an SD

card library. If you're holding out with an old version of the Arduino IDE, you can [download our library here \(https://adafru.it/aP6\)](https://adafru.it/aP6). Click the **Download ZIP** button, uncompress the archive and rename the resulting folder to “**SD**”. Move this folder into your Arduino libraries folder (usually [home]/Documents/Arduino/Libraries) and restart the IDE.

Wire up the TFT as we have before (see previous page) and then wire up the microSD card breakout.

On an Uno/Duemilanove you will want to use the following connections: Connect the SD card with **DI** going to pin **11**, **DO** going to pin **12** and **SCK** going to pin **13** (this is standard on all our shields) Then pin **10** goes to **CS**. For MEGA, check the SPI connections go to the Mega's SPI pins (not 10-13)

Copy the **woof.bmp** and **miniwoof.bmp** files to a microSD card and insert it into the breakout. Run the **tftbmp** example sketch in the TFTLCD library, you should see the images show up. If not, check the serial monitor for hints as to why it may not be working.



Downloads

This tutorial is for the original 2.8" TFT breakout, we have an updated tutorial now at <https://learn.adafruit.com/adafruit-2-dot-8-color-tft-touchscreen-breakout->

v2 for the most recent version of this product, but are keeping this tutorial for historical reference

- [Touch screen library](https://adafru.it/aT1) (<https://adafru.it/aT1>)
- [TFT LCD library](https://adafru.it/aHk) (<https://adafru.it/aHk>)
- [GFX library](https://adafru.it/aJa) (<https://adafru.it/aJa>)
- [SD card library](https://adafru.it/aP6) (<https://adafru.it/aP6>) (**not needed** if running [up-to-date Arduino IDE](#) (<https://adafru.it/aHs>)!)
- [Breakout board schematic](https://adafru.it/cmo) (<https://adafru.it/cmo>)

Support Forums

[Support Forums](https://adafru.it/forums) (<https://adafru.it/forums>)