



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



□

Adafruit 3.5" 320x480 Color TFT Touchscreen Breakout

Created by lady ada

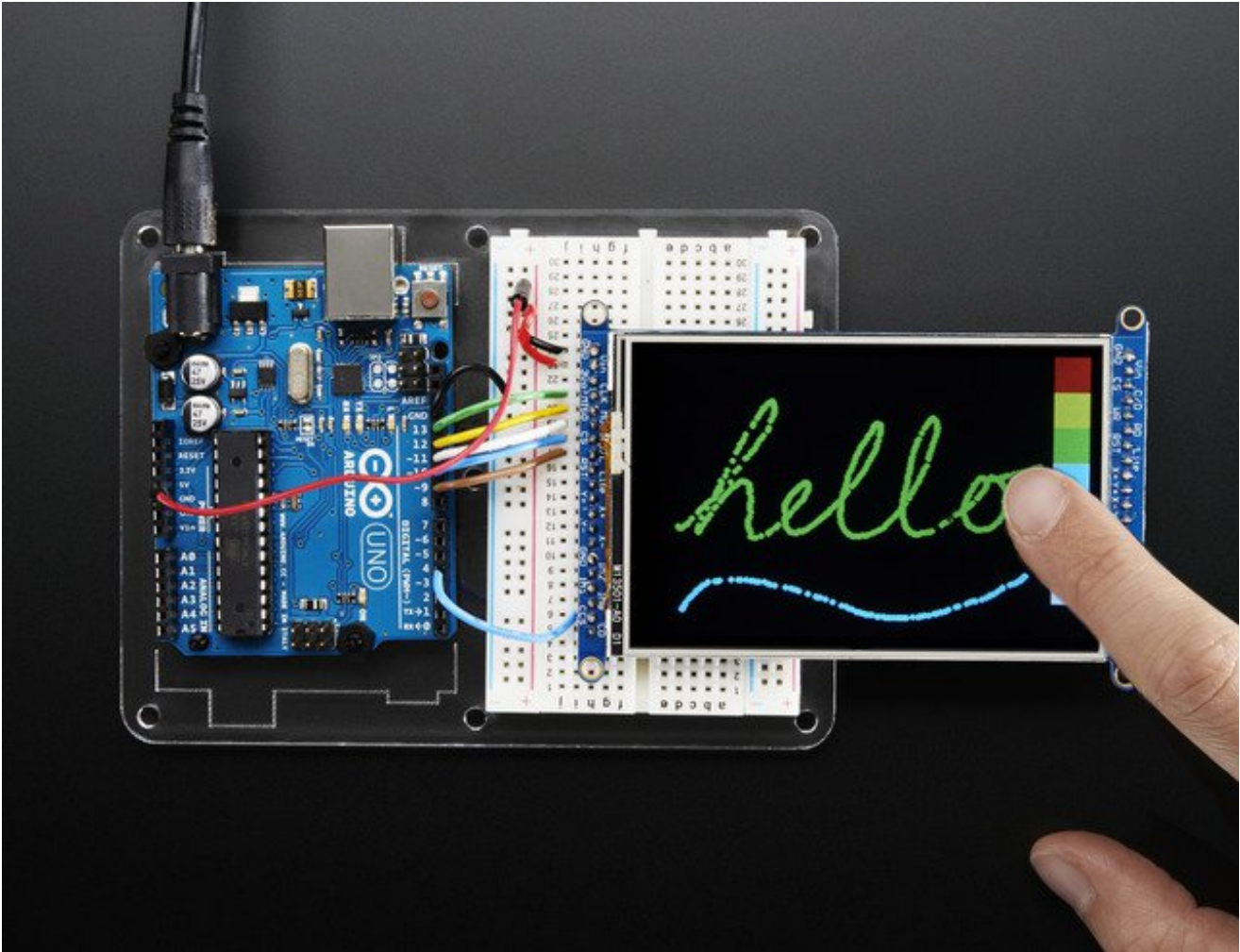


Last updated on 2017-01-30 01:59:14 AM UTC

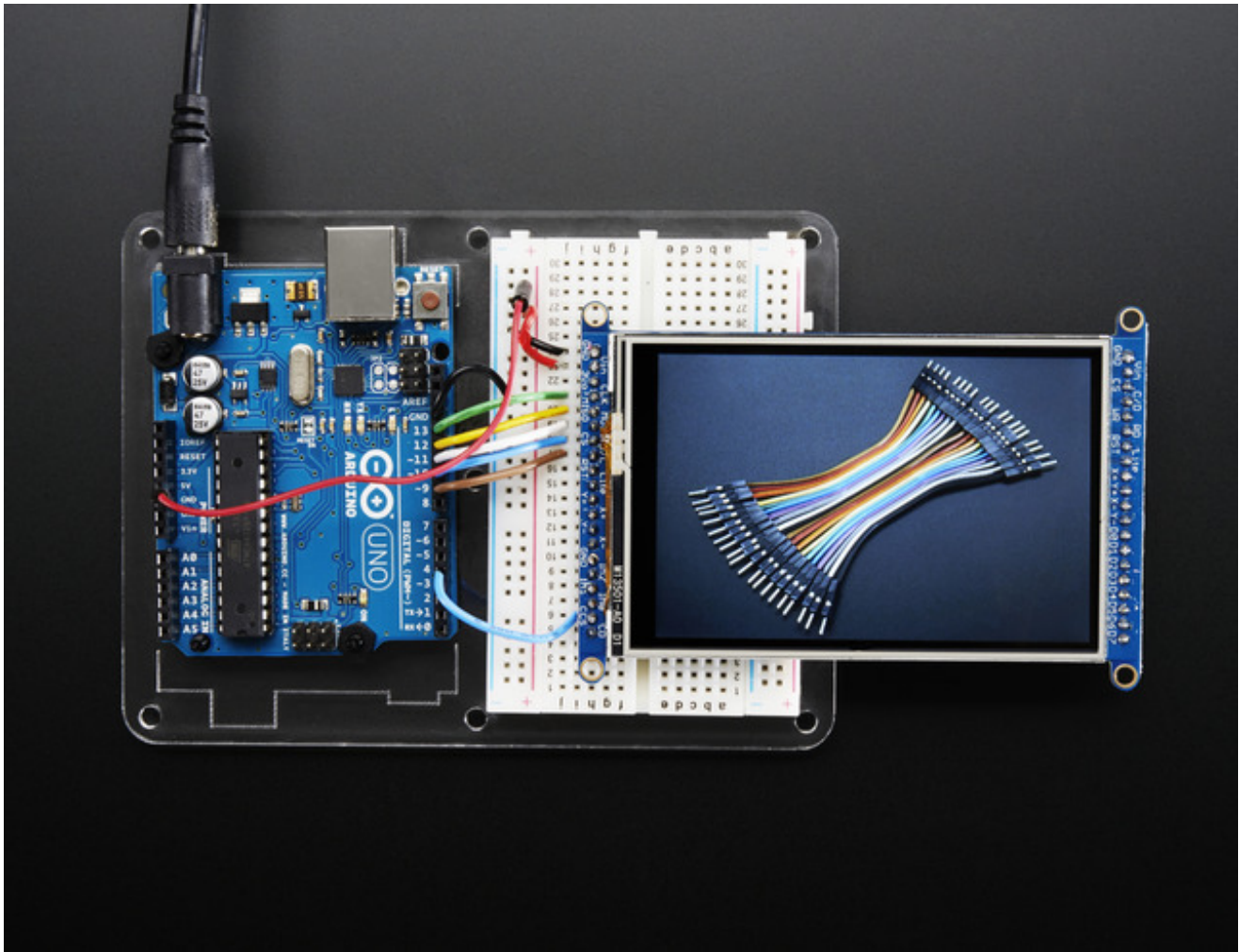
Guide Contents

Guide Contents	2
Overview	3
Pinouts	7
SPI Mode	7
8-Bit Mode	9
Wiring & Test	11
Assembling Header	11
Prepare the header strip:	12
Add the breakout board:	13
Add the breakout board:	14
8-Bit Wiring & Test	16
8-Bit Wiring	16
Part 1 - Power & backlight test	16
Part 2 - Data Bus Lines	18
8-Bit Library Install	22
Prepare TFTLCD Library	23
SPI Wiring & Test	29
SPI Mode Jumpers	29
Wiring	32
Install Adafruit HX8357 TFT Library	33
Bitmaps (SPI Mode)	36
Adafruit GFX library	39
Touchscreen	41
Download Library	42
Touchscreen Paint (SPI mode)	42
Touchscreen Paint (8-Bit mode)	44
Downloads	46
Datasheets & Files	46
Schematic and PCB Print	46

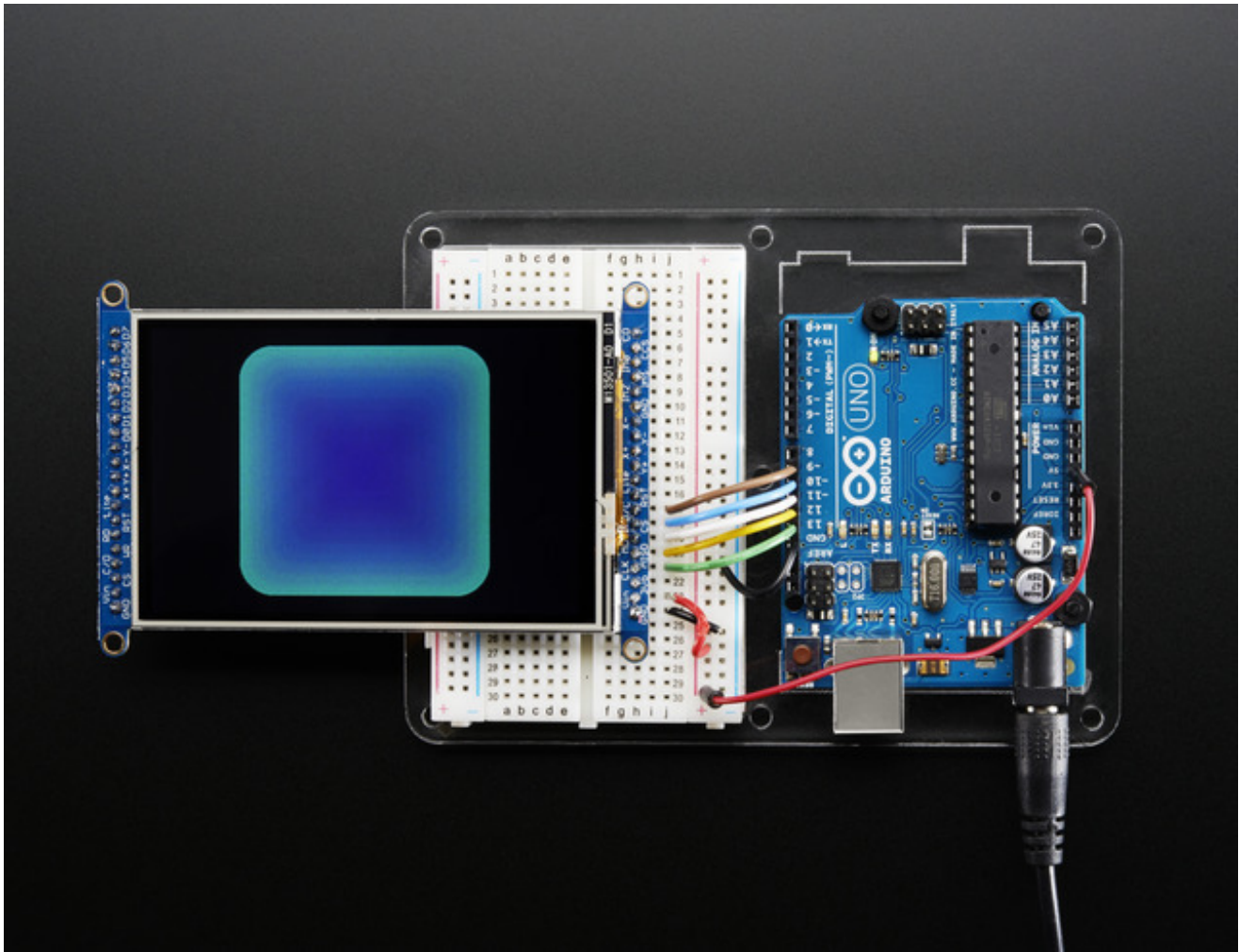
Overview



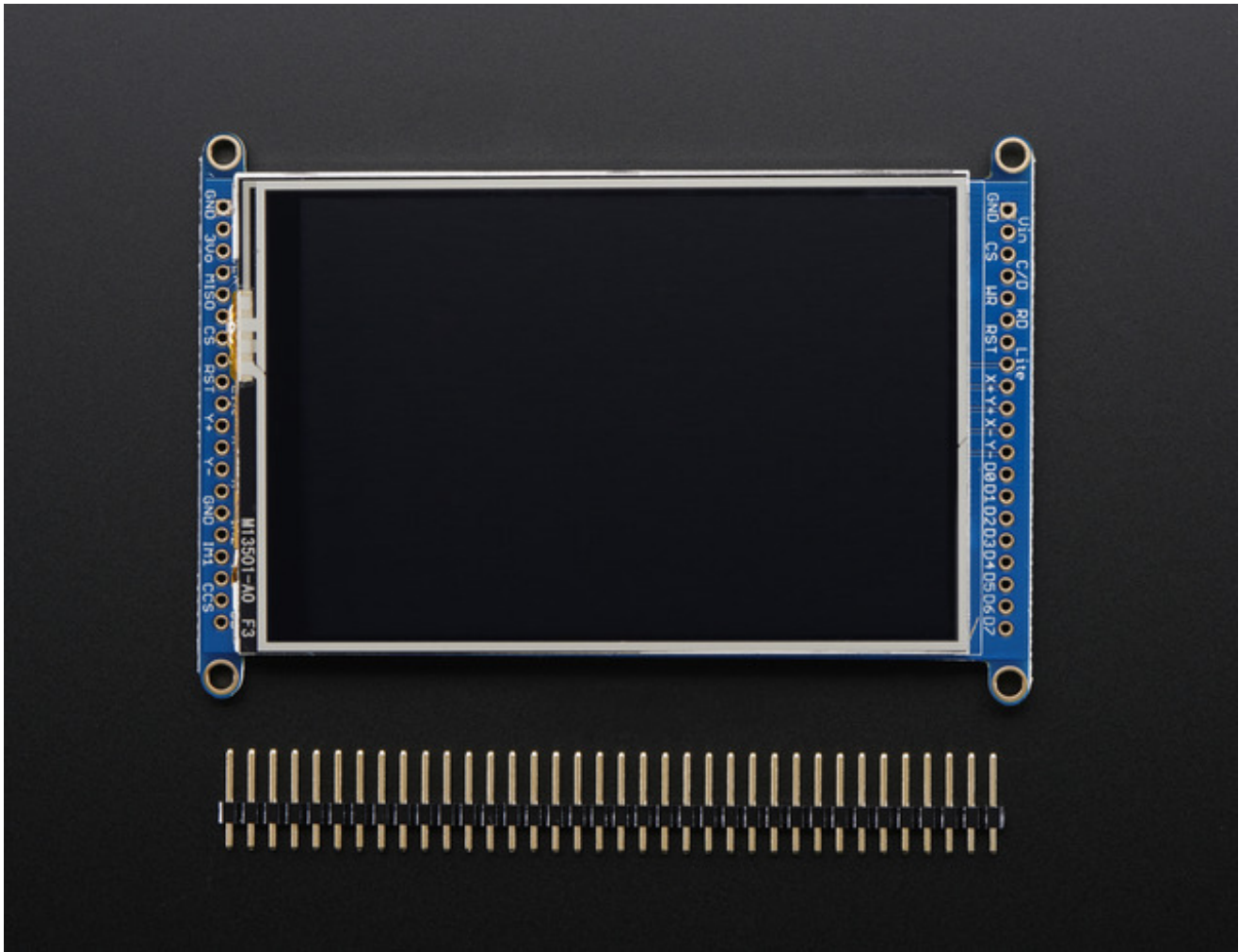
Add some jazz & pizzazz to your project with a color touchscreen LCD. This TFT display is big (3.5" diagonal) bright (6 white-LED backlight) and colorful! 480x320 pixels with individual RGB pixel control, this has way more resolution than a black and white 128x64 display, and double our 2.8" TFT. As 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. **The display can be used in two modes: 8-bit or SPI.** For 8-bit mode, you'll need 8 digital data lines and 4 or 5 digital control lines to read and write to the display (12 lines total). SPI mode requires only 5 pins total (SPI data in, data out, clock, select, and d/c) but is slower than 8-bit mode. In addition, 4 pins are required for the touch screen (2 digital, 2 analog) or [you can purchase and use our resistive touchscreen controller \(not included\) to use I2C or SPI \(http://adafru.it/1571\)](http://adafru.it/1571).

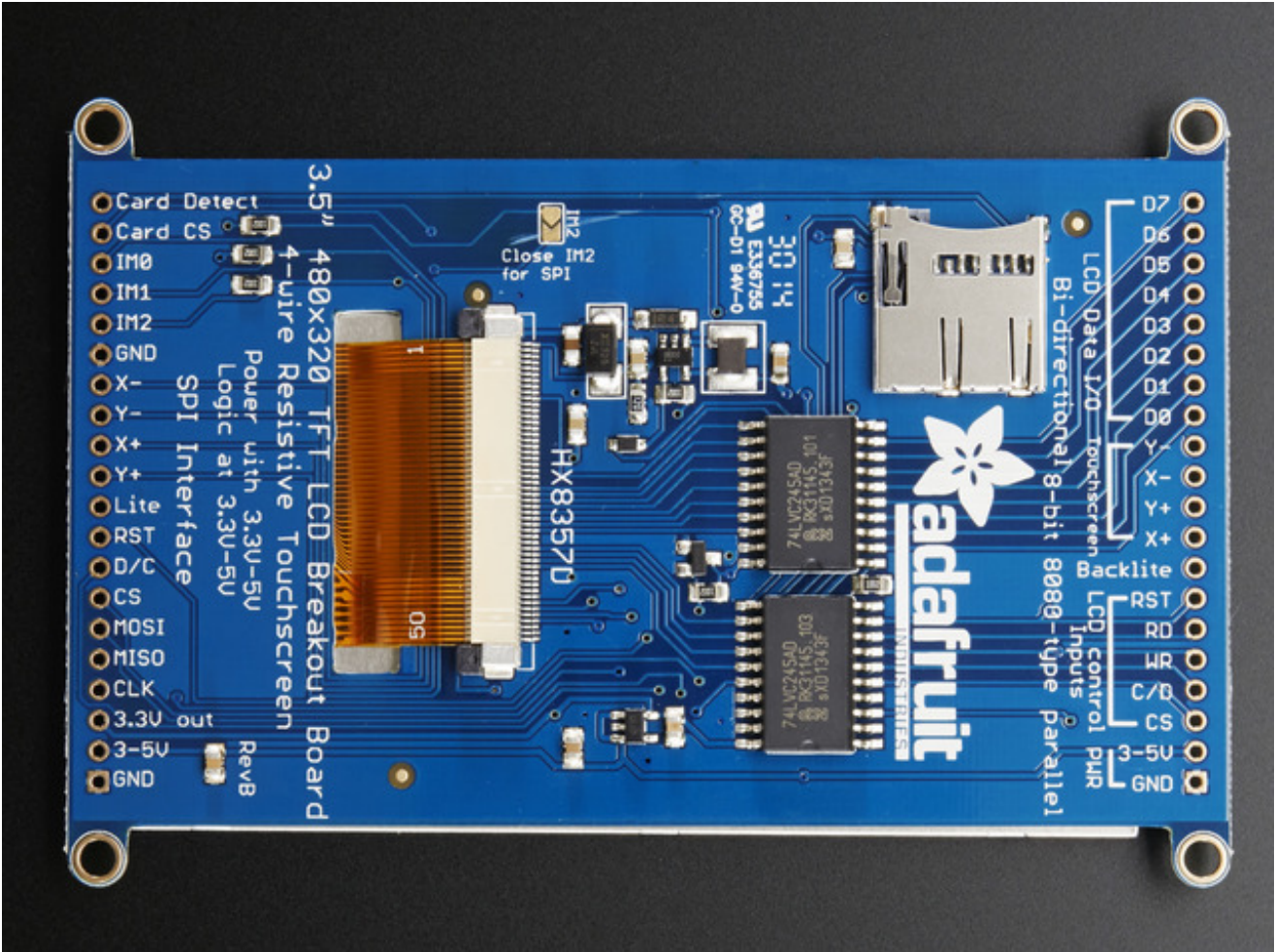


Of course, we wouldn't just leave you with a datasheet and a "good luck!". For 8-bit interface fans [we've written a full open source graphics library that can draw pixels, lines, rectangles, circles, text, and more](http://adafruit.it/aHk) (<http://adafruit.it/aHk>). For [SPI users, we have a library as well](http://adafruit.it/dQW) (<http://adafruit.it/dQW>), its separate from the 8-bit library since both versions are heavily optimized. We [also have a touch screen library that detects x, y and z \(pressure\)](http://adafruit.it/aT1) (<http://adafruit.it/aT1>) and example code to demonstrate all of it.





Pinouts

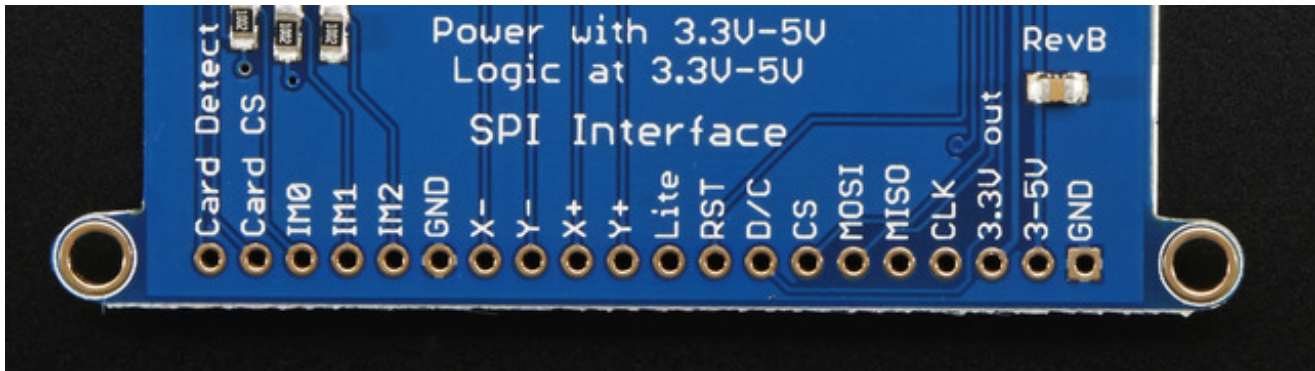


The 3.5" TFT display on this breakout supports many different modes - so many that the display itself has 50 pins. However, we think most people really only use 2 different modes, either "SPI" mode or 8-bit mode. Each 'side' of the display has all the pins required for that mode. You can switch between modes, by rewiring the display, but it cannot be used in two modes at the same time!

All logic pins, both 8-bit and SPI sides, are 3-5V logic level compatible, the 74LVX245 chips on the back perform fast level shifting so you can use either kind of logic levels. If there's data output, the levels are at at 3.3V

SPI Mode

This is what we think will be a popular mode when speed is not of the utmost importance. It doesn't use as many pins (only 4 to draw on the TFT if you skip the MISO pin), is fairly flexible, and easy to port to various microcontrollers. It also allows using a microSD card socket on the same SPI bus. However, its slower than parallel 8-bit mode because you have to send each bit at a time instead of 8-bits at a time. Tradeoffs!

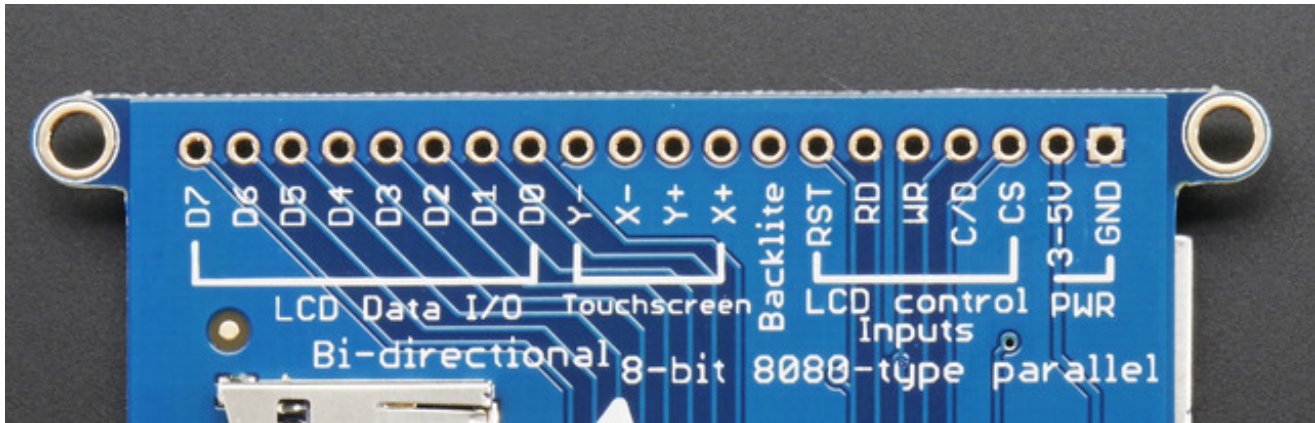


- **GND** - this is the power and signal ground pin
- **3-5V / Vin** - this is the power pin, connect to 3-5VDC - it has reverse polarity protection but try to wire it right!
- **3.3Vout** - this is the 3.3V output from the onboard regulator
- **CLK** - this is the SPI clock input pin
- **MISO** - this is the SPI Master In Slave Out pin, its used for the SD card mostly, and for debugging the TFT display. It isn't necessary for using the TFT display which is write-only
- **MOSI** - this is the SPI Master Out Slave In pin, it is used to send data from the microcontroller to the SD card and/or TFT
- **CS** - this is the TFT SPI chip select pin
- **D/C** - this is the TFT SPI data or command selector pin
- **RST** - this is the TFT reset pin. There's auto-reset circuitry on the breakout so this pin is not required but it can be helpful sometimes to reset the TFT if your setup is not always resetting cleanly. Connect to ground to reset the TFT
- **Lite** - this is the PWM input for the backlight control. It is by default pulled high (backlight on) you can PWM at any frequency or pull down to turn the backlight off
- **Y+ X+ Y- X-** these are the 4 resistive touch screen pads, which can be read with analog pins to determine touch points. They are completely separated from the TFT electrically (the overlay is glued on top)
- **IM2 IM1 IM0** - these are interface control set pins. In general these breakouts aren't used, and instead the onboard jumpers are used to fix the interface to SPI or 8-bit. However, we break these out for advanced use and also for our test procedures
- **Card CS / CCS** - this is the SD card chip select, used if you want to read from the SD card.
- **Card Detect / CD** - this is the SD card detect pin, it floats when a card is inserted,

and tied to ground when the card is not inserted. We don't use this in our code but you can use this as a switch to detect if an SD card is in place without trying to electrically query it. Don't forget to use a pullup on this pin if so!

8-Bit Mode

This mode is for when you have lots of pins and want more speed. In this mode we send 8 bits at a time, so it needs way more pins, 12 or so (8 bits plus 4 control)! If your microcontroller



- **GND** - this is the power and signal ground pin
- **3-5V (Vin)**- this is the power pin, connect to 3-5VDC - it has reverse polarity protection but try to wire it right!
- **CS** - this is the TFT 8-bit chip select pin (it is also tied to the SPI mode **CS** pin)
- **C/D** - this is the TFT 8-bit data or command selector pin (it is also tied to the SPI mode **C/D** pin)
- **WR** - this is the TFT 8-bit write strobe pin. It is also connected to the **SPI CLK** pin
- **RD** - this is the TFT 8-bit read strobe pin. You may not need this pin if you don't want to read data from the display
- **RST** - this is the TFT reset pin. There's auto-reset circuitry on the breakout so this pin is not required but it can be helpful sometimes to reset the TFT if your setup is not always resetting cleanly. Connect to ground to reset the TFT
- **Backlite** - this is the PWM input for the backlight control. It is by default pulled high (backlight on) you can PWM at any frequency or pull down to turn the backlight off
- **Y+ X+ Y- X-** these are the 4 resistive touch screen pads, which can be read with analog pins to determine touch points. They are completely separated from the TFT electrically (the overlay is glued on top)
- **D0** thru **D7** - these are the 8 bits of parallel data sent to the TFT in 8-bit mode **D0** is the least-significant-bit and **D7** is the MSB



Wiring & Test

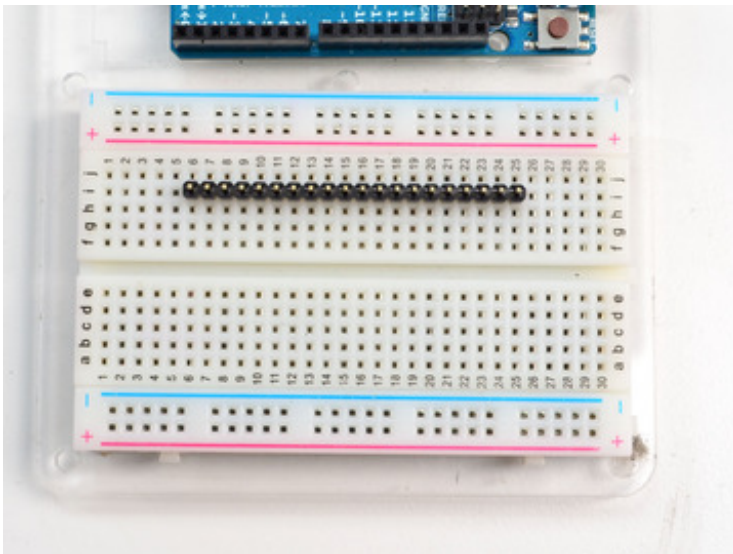
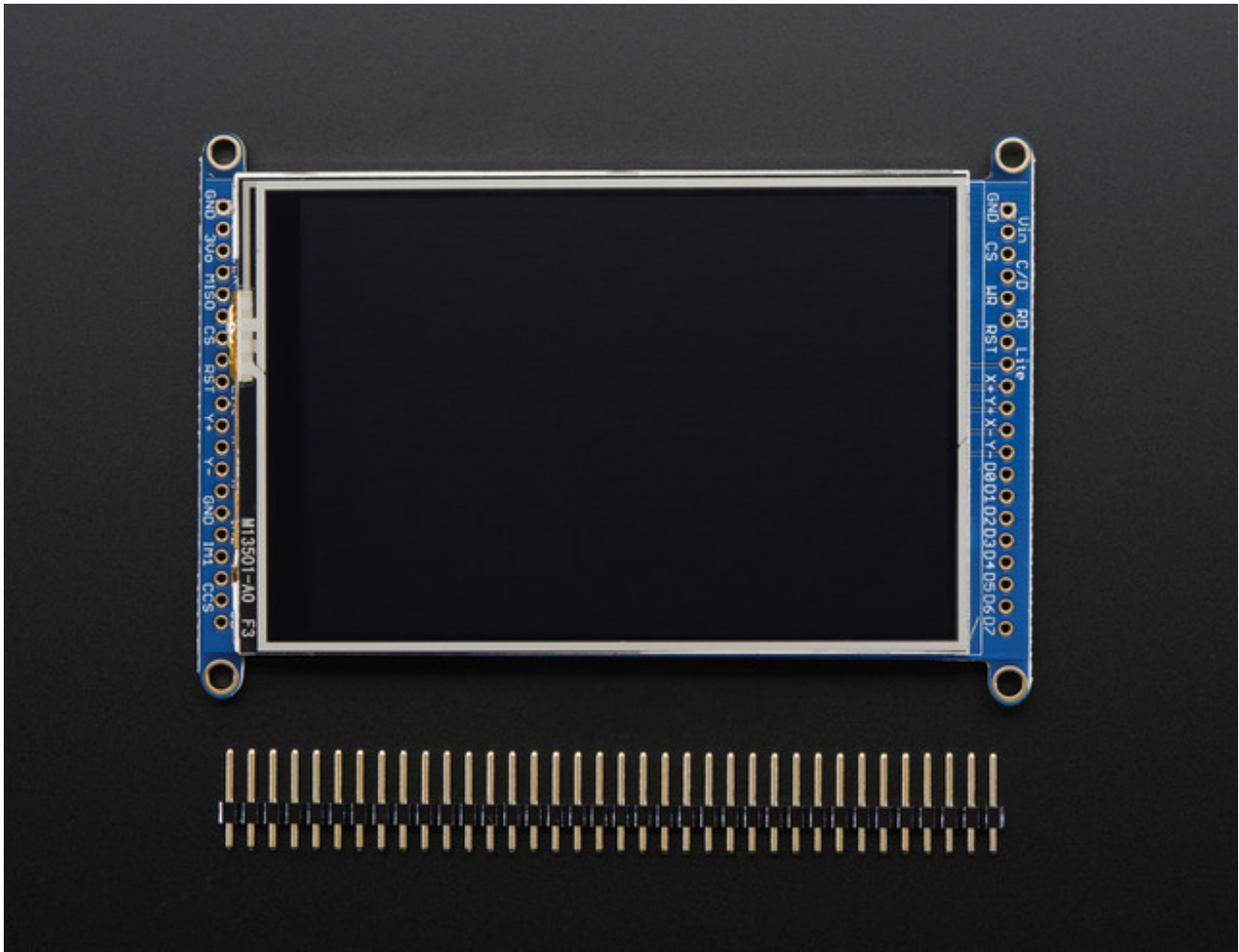
We tried to make this TFT breakout useful for both high-pin microcontrollers that can handle 8-bit data transfer modes as well as low-pincount micros like the Arduino UNO and Leonardo that are OK with SPI.

Essentially, the tradeoff is pins for speed. SPI is about 2-4 times slower than 8-bit mode, but that may not matter for basic graphics!

In addition, SPI mode has the benefit of being able to use the onboard microSD card socket for reading images. We don't have support for this in 8-bit mode so if you want to have an all-in-one image viewer type application, use SPI!

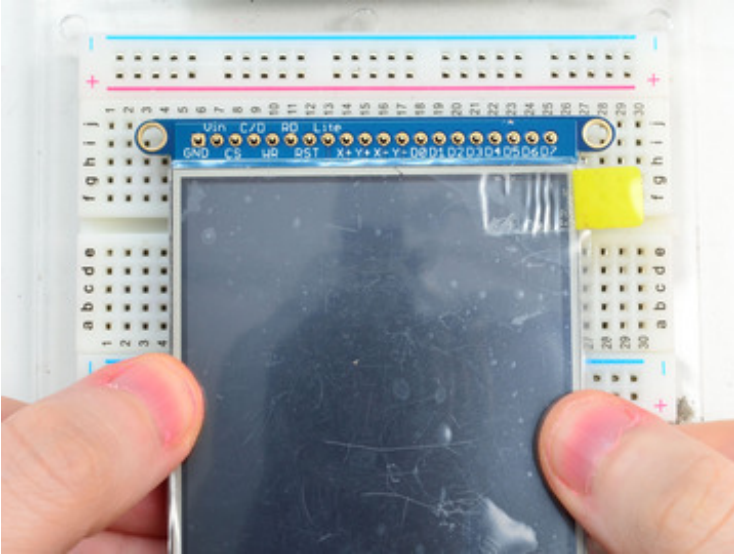
Assembling Header

Either way, if you're using a breadboard, you'll need to solder header onto one or two of the sides. The procedure is the same for both sides



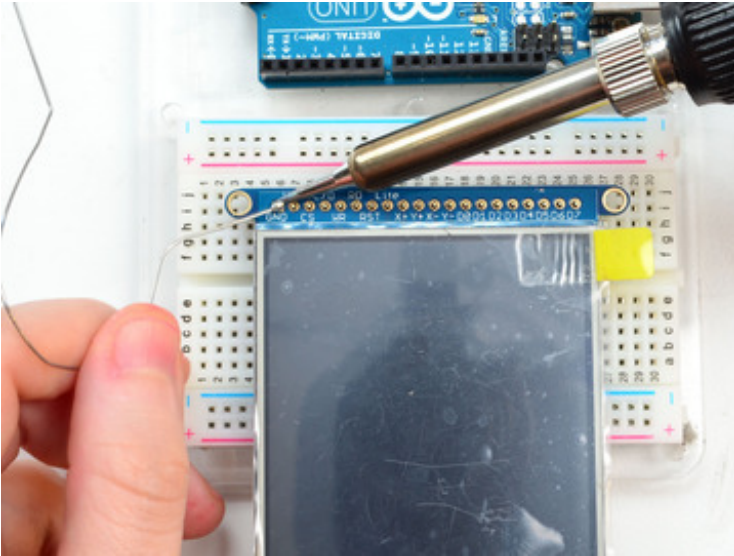
Prepare the header strip:

Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**



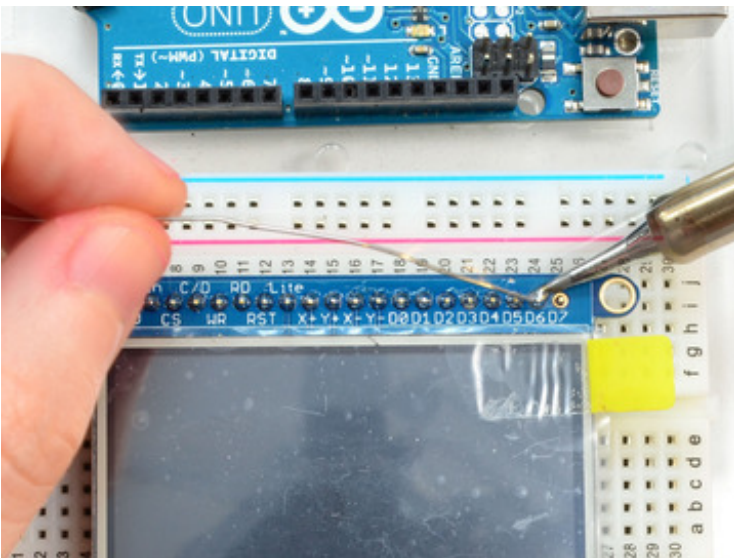
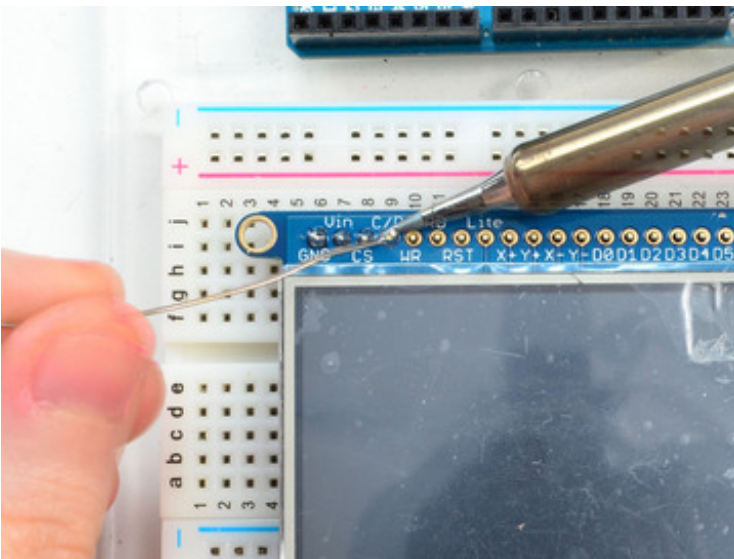
Add the breakout board:

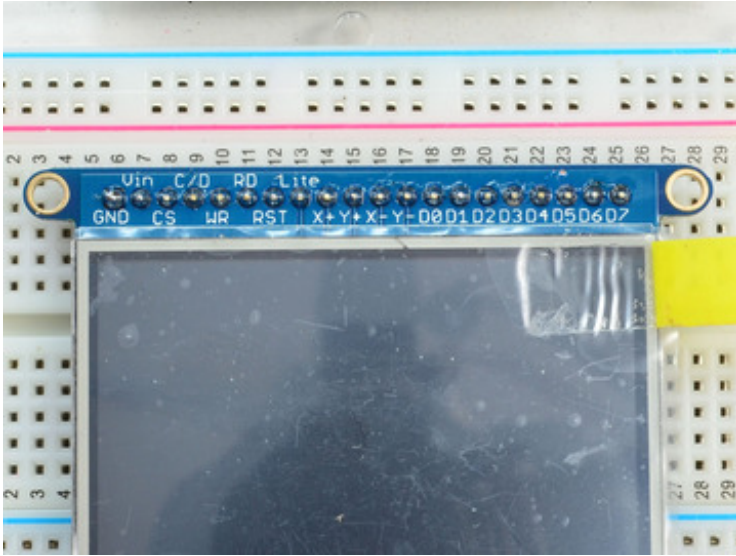
Place the breakout board over the pins so that the short pins poke through the breakout pads



Add the breakout board:

Place the breakout board over the pins so that the short pins poke through the breakout pads





You're done! Check your solder joints visually and continue onto the next steps

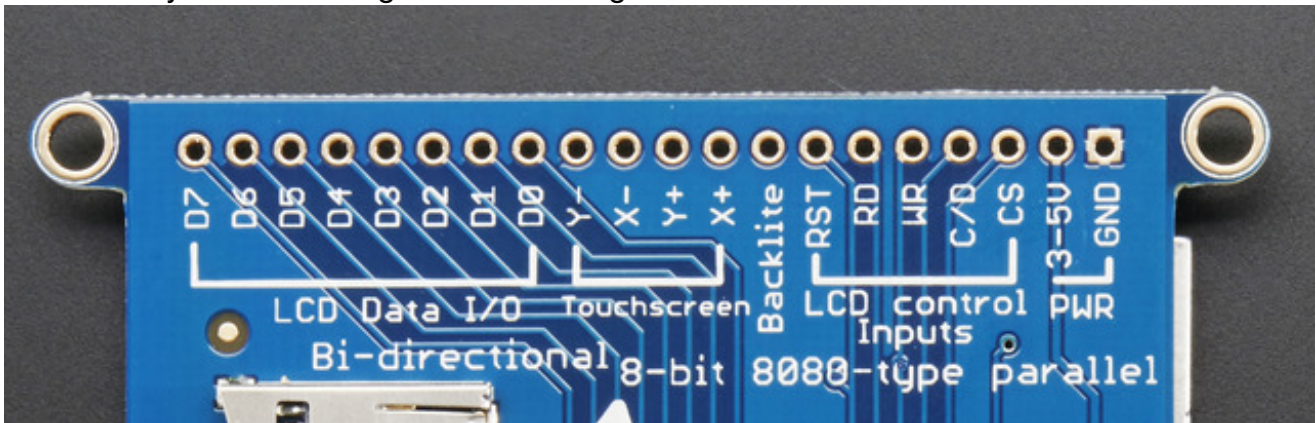
8-Bit Wiring & Test

8-Bit Wiring

Wiring up the 8-bit mode is kind of a pain, so we really only recommend doing it for UNO (which we show) and Mega (which we describe, and is pretty easy since its 8 pins in a row). Anything else, like a Leonardo or Micro, we strongly recommend going with SPI mode since we don't have an example for that. Any other kind of 'Arduino compatible' that isn't an Uno, try SPI first. The 8-bit mode is hand-tweaked in the **Adafruit_TFTLCD pin_magic.h** file. Its really only for advanced users who are totally cool with figuring out bitmasks for various ports & pins.

Really, we'll show how to do the UNO but anything else? go with SPI!

Make sure you're soldering and connecting to the 8-bit side!

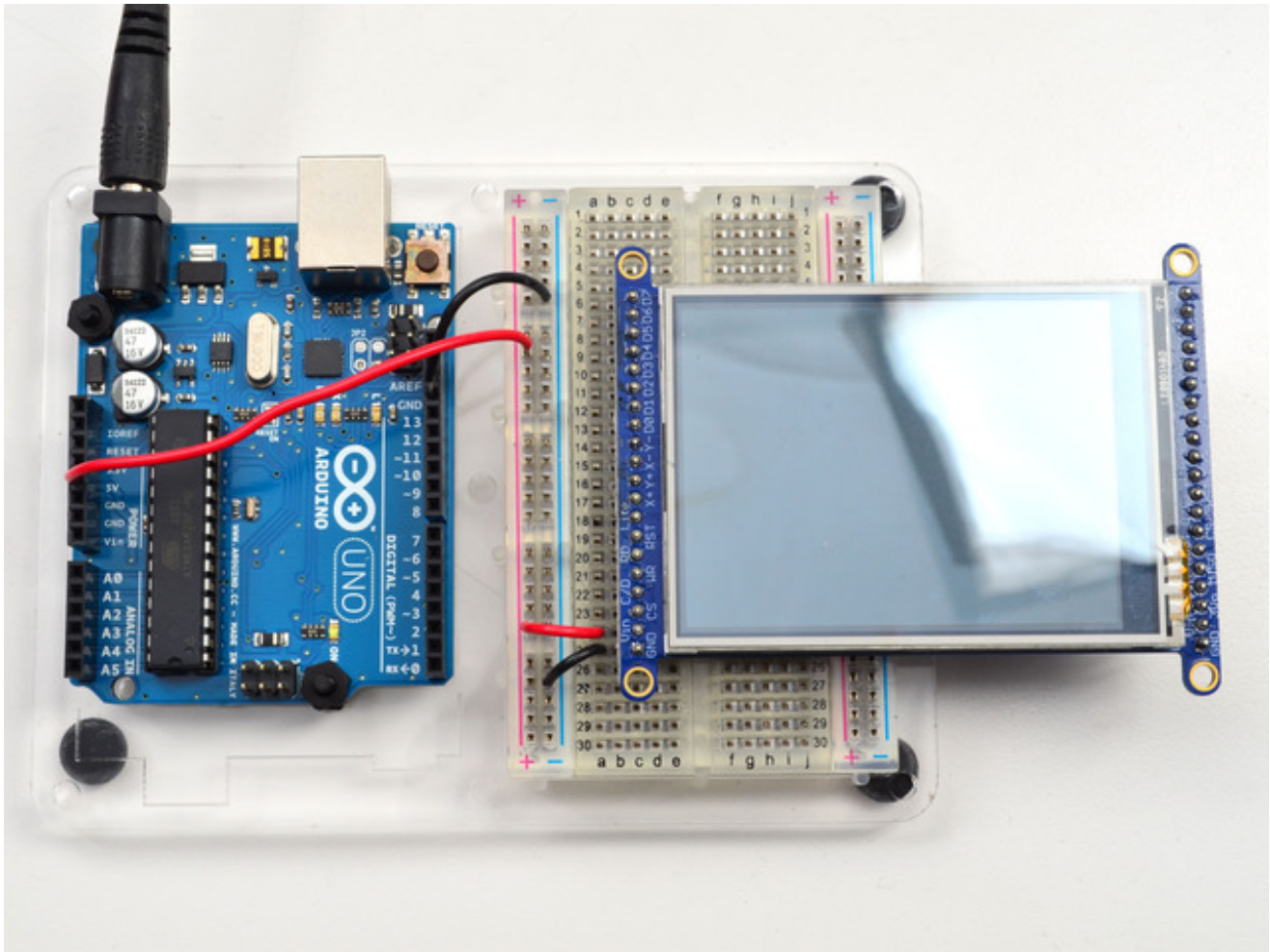


Part 1 - Power & backlight test

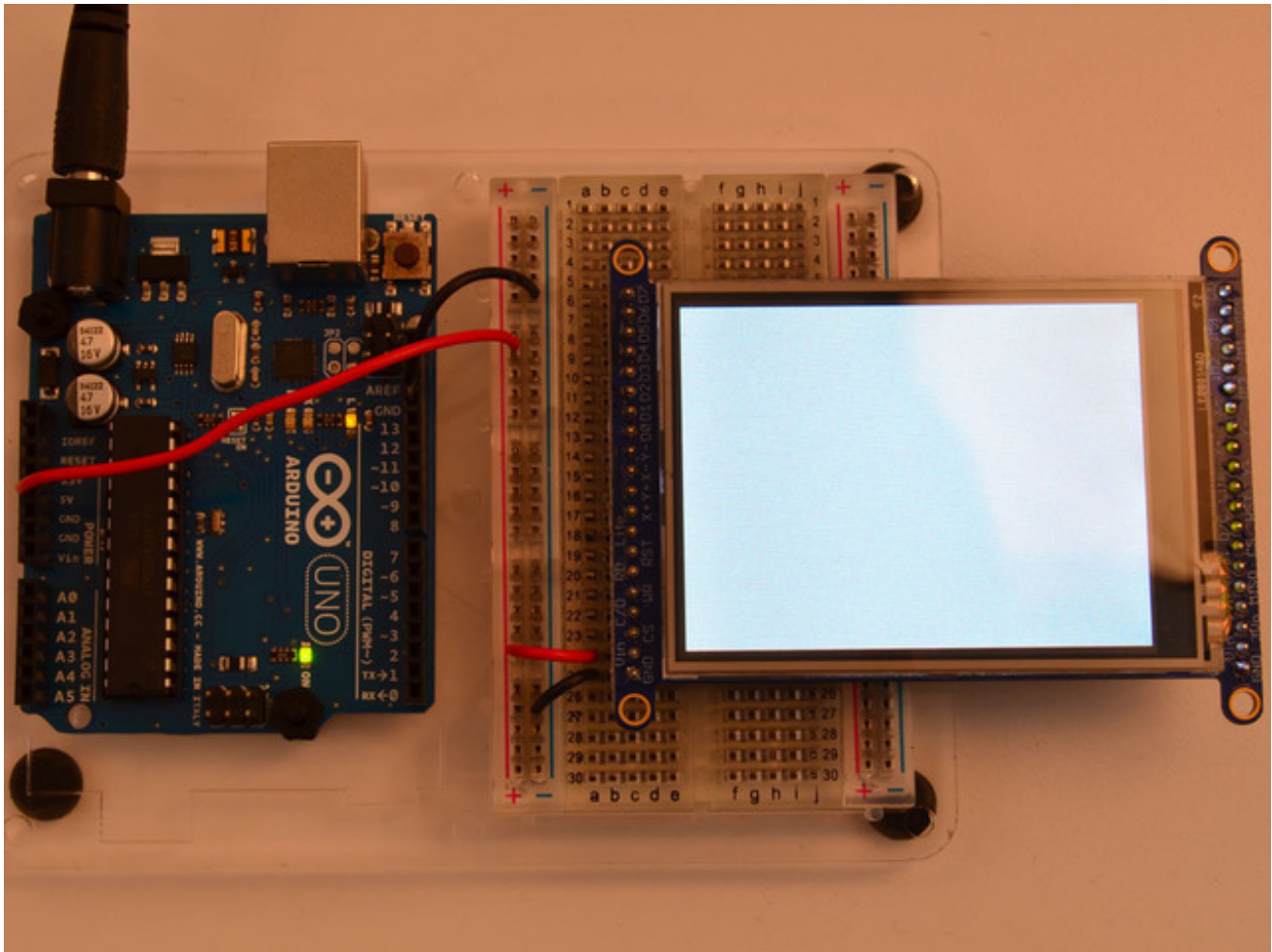
In these images we show using our 2.8" TFT but its the exact same pinout, just a tad smaller!

Begin by wiring up the **3-5VDC** and **GND** pins.

Connect the **3-5V** pin to **5V** and **GND** to **GND** on your Arduino. I'm using the breadboard rails but you can also just wire directly.



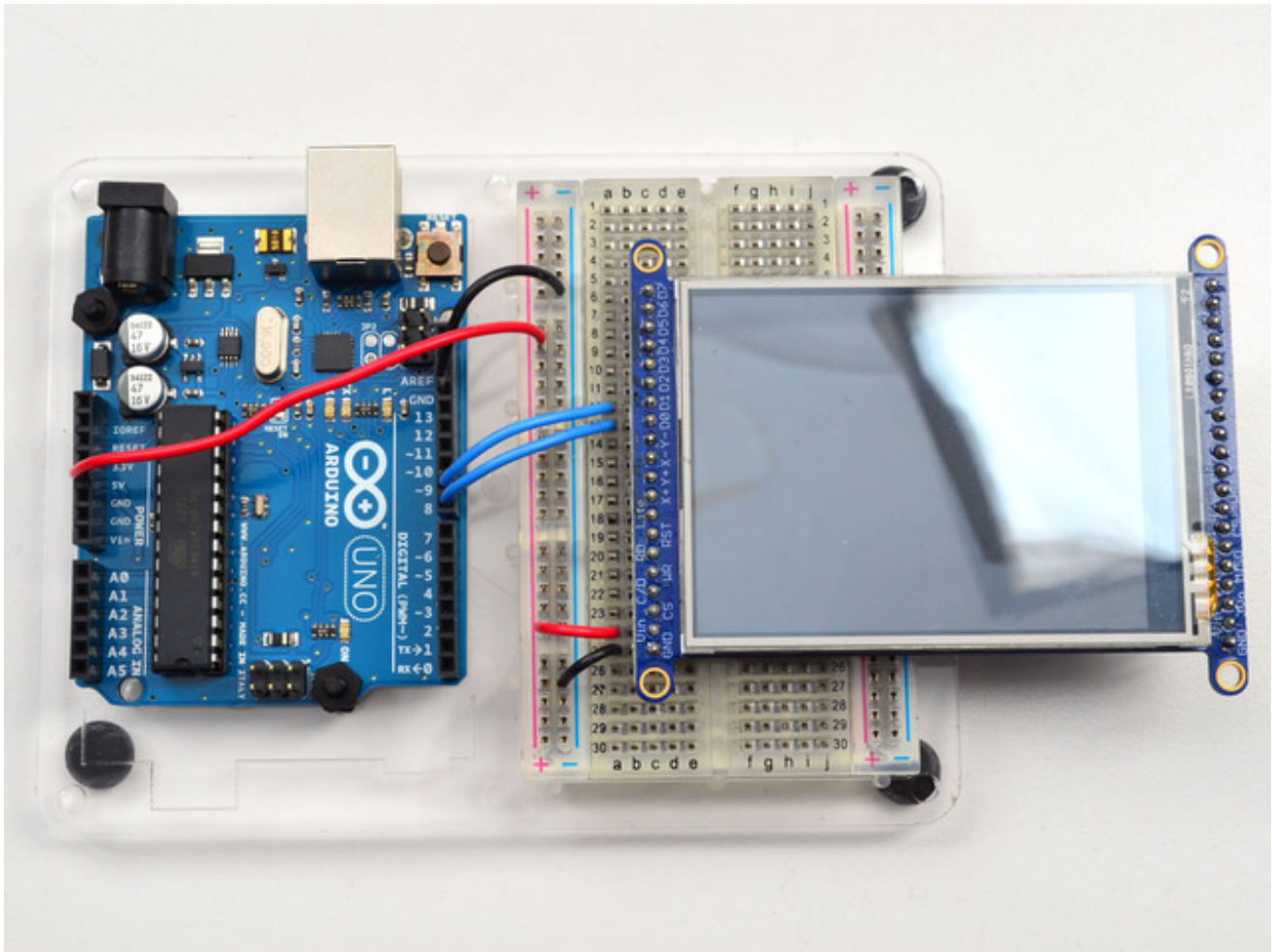
Power it up and you should see the white backlight come on



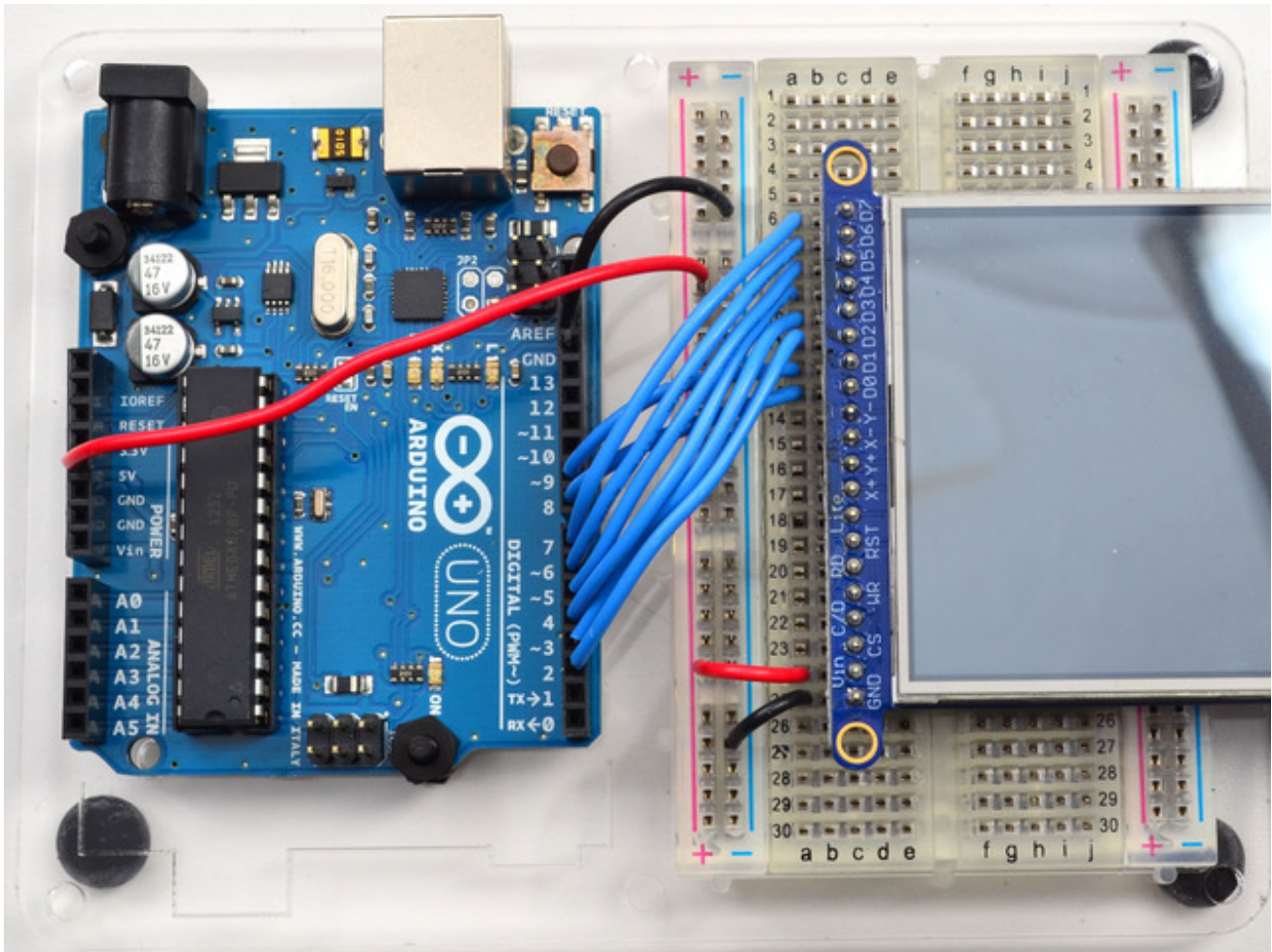
Part 2 - Data Bus Lines

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.

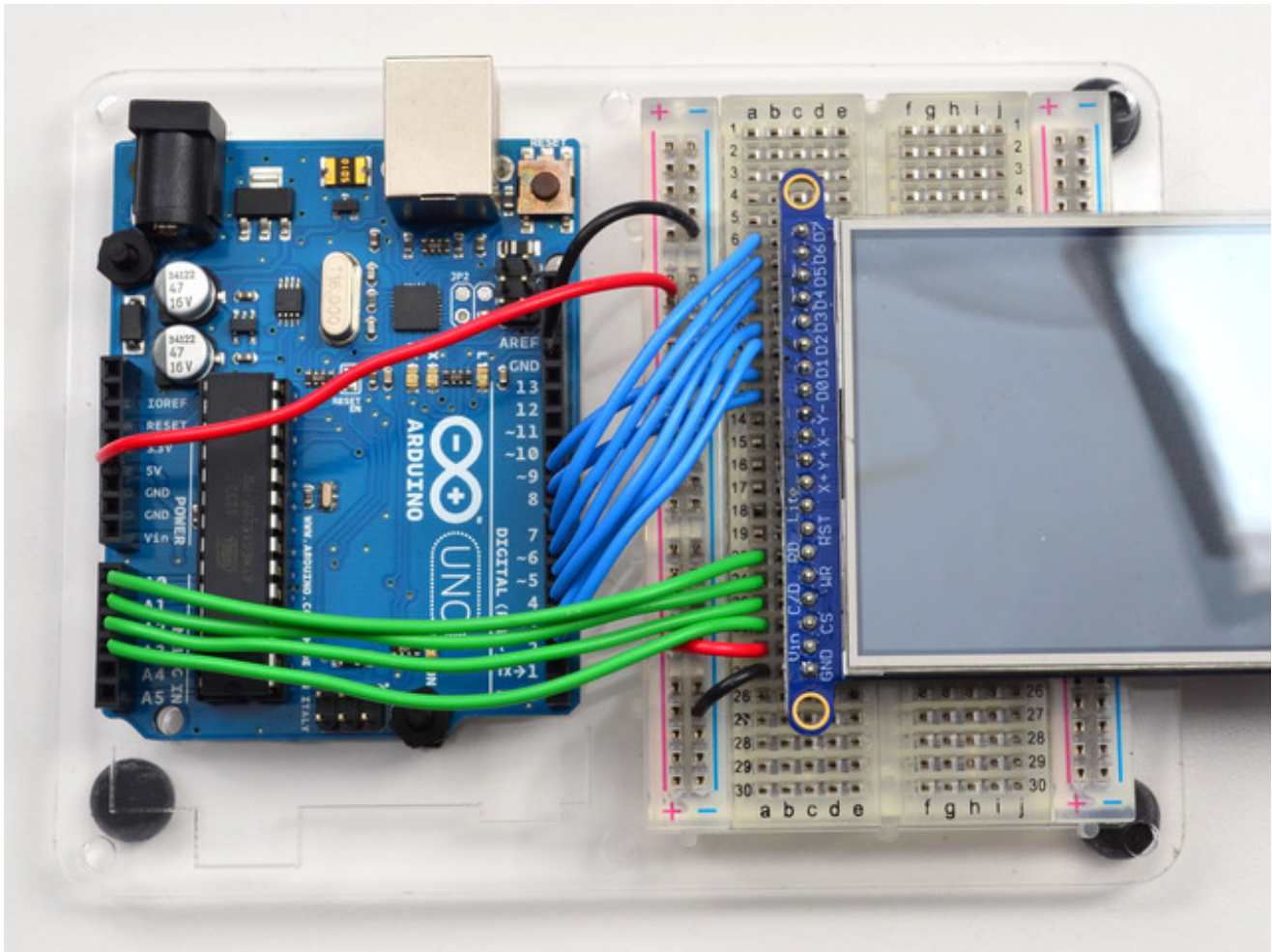
However, they are not in that order! D0 and D1 go to **digital #8** and **#9**, then D2-D7 connect to **#2** thru **#7**. This is because Arduino pins #0 and #1 are used for serial data so we can't use them



Begin by connecting **D0** and **D1** to digital **#8** and **9** respectively as seen above. If you're using a Mega, connect the TFT Data Pins **D0-D1** to Mega pins **#22-23**, in that order. Those Mega pins are on the 'double' header.

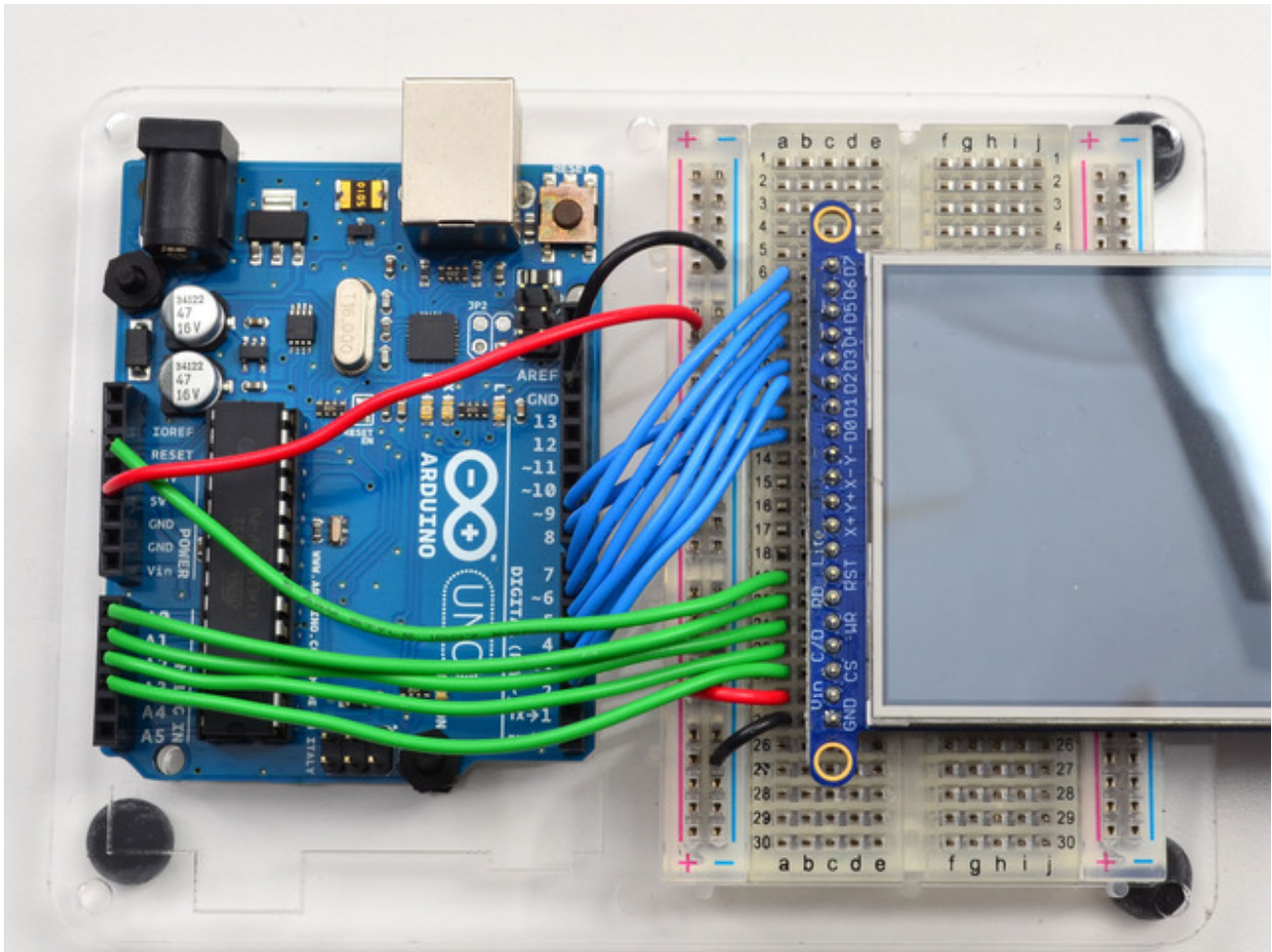


Now you can connect the remaining 6 pins over. Connect **D2-D7** on the TFT pins to digital **2** thru **7** in that order. If you're using a Mega, connect the TFT Data Pins **D2-D7** to Mega pins **#24-29**, in that order. Those Mega pins are on the 'double' header.



In addition to the 8 data lines, you'll also need 4 or 5 control lines. These can later be reassigned to any digital pins, they're just what we have in the tutorial by default.

- 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



You can connect the seventh pin **RST (Reset)** to the Arduino Reset line if you'd like. This will reset the panel when the Arduino is Reset. You can also use a digital pin for the LCD reset if you want to manually reset. There's auto-reset circuitry on the board so you probably don't need to use this pin at all and leave it disconnected

The **RD** pin is used to read the chip ID off the TFT. Later, once you get it all working, you can remove this pin and the ID test, although we suggest keeping it since its useful for debugging your wiring.

OK! Now we can run some code

8-Bit Library Install

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: first is the [Adafruit_TFTLCD](#)

[library \(http://adafru.it/aHk\)](http://adafru.it/aHk) (this contains the low-level code specific to this device), and second is the [Adafruit GFX Library \(http://adafru.it/aJa\)](http://adafru.it/aJa) (which handles graphics operations common to many displays we carry). If you have **Adafruit_GFX** already, make sure its the most recent version since we've made updates for better performance

[Download Adafruit_TFTLCD Library](http://adafru.it/dcW)

<http://adafru.it/dcW>

[Download Adafruit_GFX Library](http://adafru.it/cBB)

<http://adafru.it/cBB>

Download both ZIP files, uncompress and rename the folders to **Adafruit_TFTLCD** (contains **Adafruit_TFTLCD.cpp** and **.h**) and **Adafruit_GFX** (contains **Adafruit_GFX.cpp** and **.h**) respectively. Then place them inside your **Arduinolibraries** folder and restart the Arduino IDE. If this is all unfamiliar, we have a [tutorial introducing Arduino library concepts and installation \(http://adafru.it/aYM\)](http://adafru.it/aYM).

Prepare TFTLCD Library

In the **Adafruit_TFTLCD** Library folder, you may need to edit **Adafruit_TFTLCD.h**. On about line 12, you will see

```
#define USE_ADAFRUIT_SHIELD_PINOUT
```

Make sure this line is commented out with `a//` in front (it should but if you're having issues, its worth checking).

Next up, we originally designed this library for 320x240 TFTs. Since this is a 480x320, we have to adjust the size the library is expecting. Open up **Adafruit_TFTLCD.cpp** and find these lines:


```
Adafruit_TFTLCD.cpp
#include "pins_arduino.h"
#include "wiring_private.h"
#include "Adafruit_TFTLCD.h"
#include "pin_magic.h"

// #define TFTWIDTH 320
// #define TFTHEIGHT 480

#define TFTWIDTH 240
#define TFTHEIGHT 320

// LCD controller chip identifiers
#define ID_932X 0
#define ID_7575 1
#define ID_9341 2
#define ID_HX8357D 3
#define ID_UNKNOWN 0xFF

#include "registers.h"

// Constructor for breakout board (configurable LCD control lines).
// Can still use this w/ shield, but parameters are ignored
Raw: T-----XEmacs: Adafruit_TFTLCD.cpp (C++/1 PenDel Font Abbrev) -----L21--2%--
```

Comment out the 240 and 320 lines, and uncomment the 320 and 480 lines:

```
Adafruit_TFTLCD.cpp
#include "pins_arduino.h"
#include "wiring_private.h"
#include "Adafruit_TFTLCD.h"
#include "pin_magic.h"

#define TFTWIDTH 320
#define TFTHEIGHT 480

// #define TFTWIDTH 240
// #define TFTHEIGHT 320

// LCD controller chip identifiers
#define ID_932X 0
#define ID_7575 1
#define ID_9341 2
#define ID_HX8357D 3
#define ID_UNKNOWN 0xFF

#include "registers.h"

// Constructor for breakout board (configurable LCD control lines).
// Can still use this w/ shield, but parameters are ignored.
Raw:T-----XEmacs: Adafruit_TFTLCD.cpp (C++/1 PenDel Font Abbrev) -----L34--2%
```

Save it, now you can upload the demo!

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.