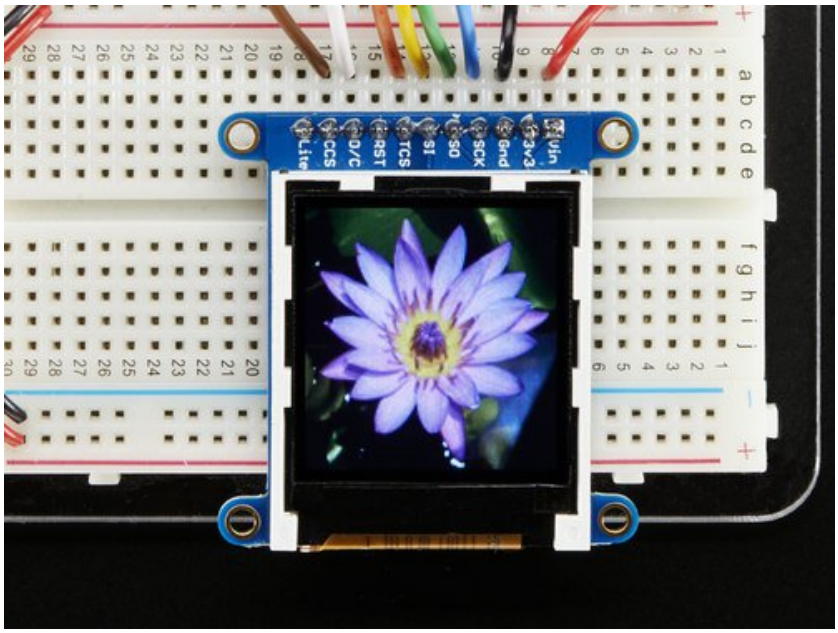
Adafruit 1.44" Color TFT with Micro SD Socket

Created by lady ada



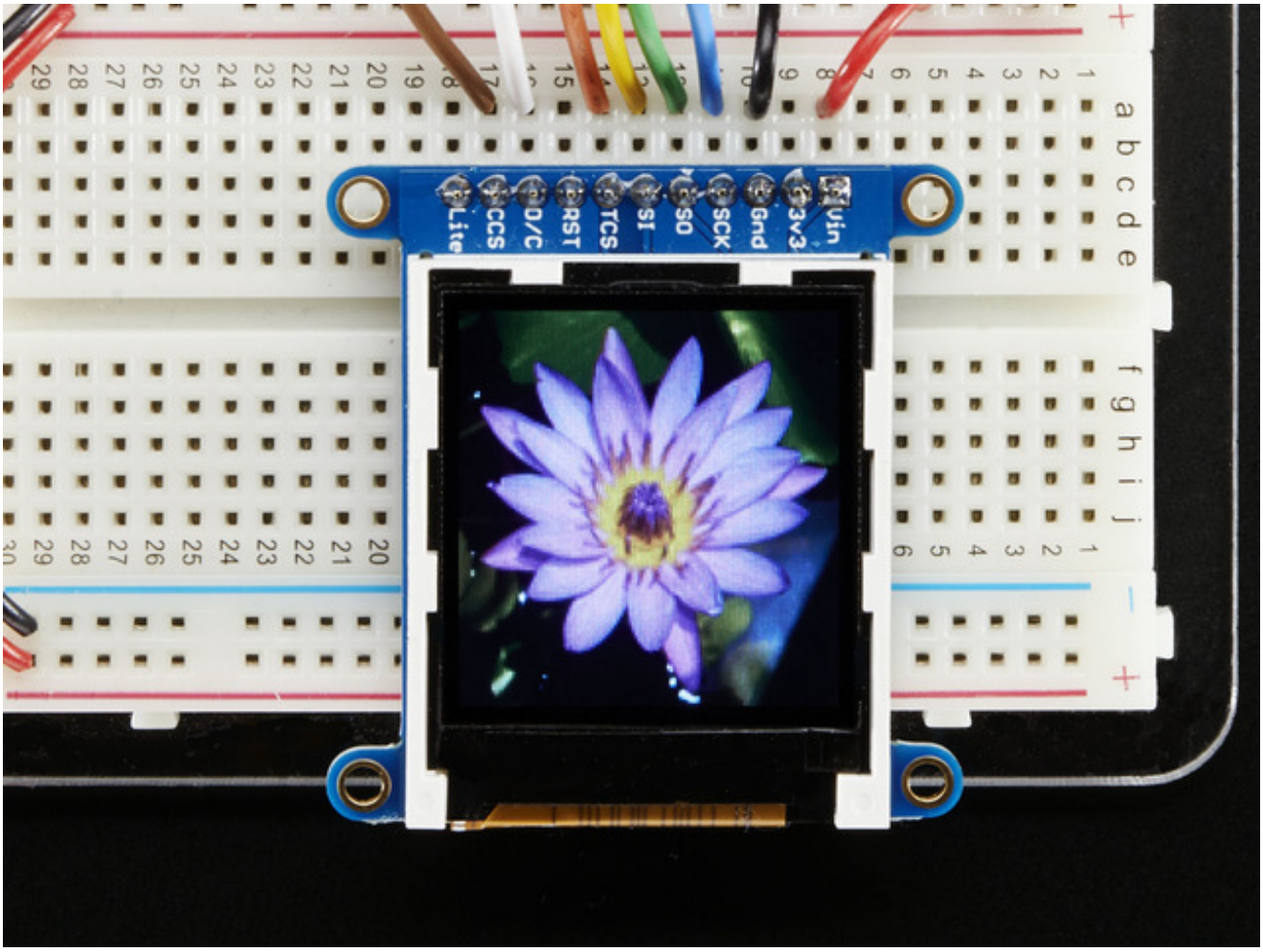
Last updated on 2016-08-03 04:01:07 AM UTC

Guide Contents

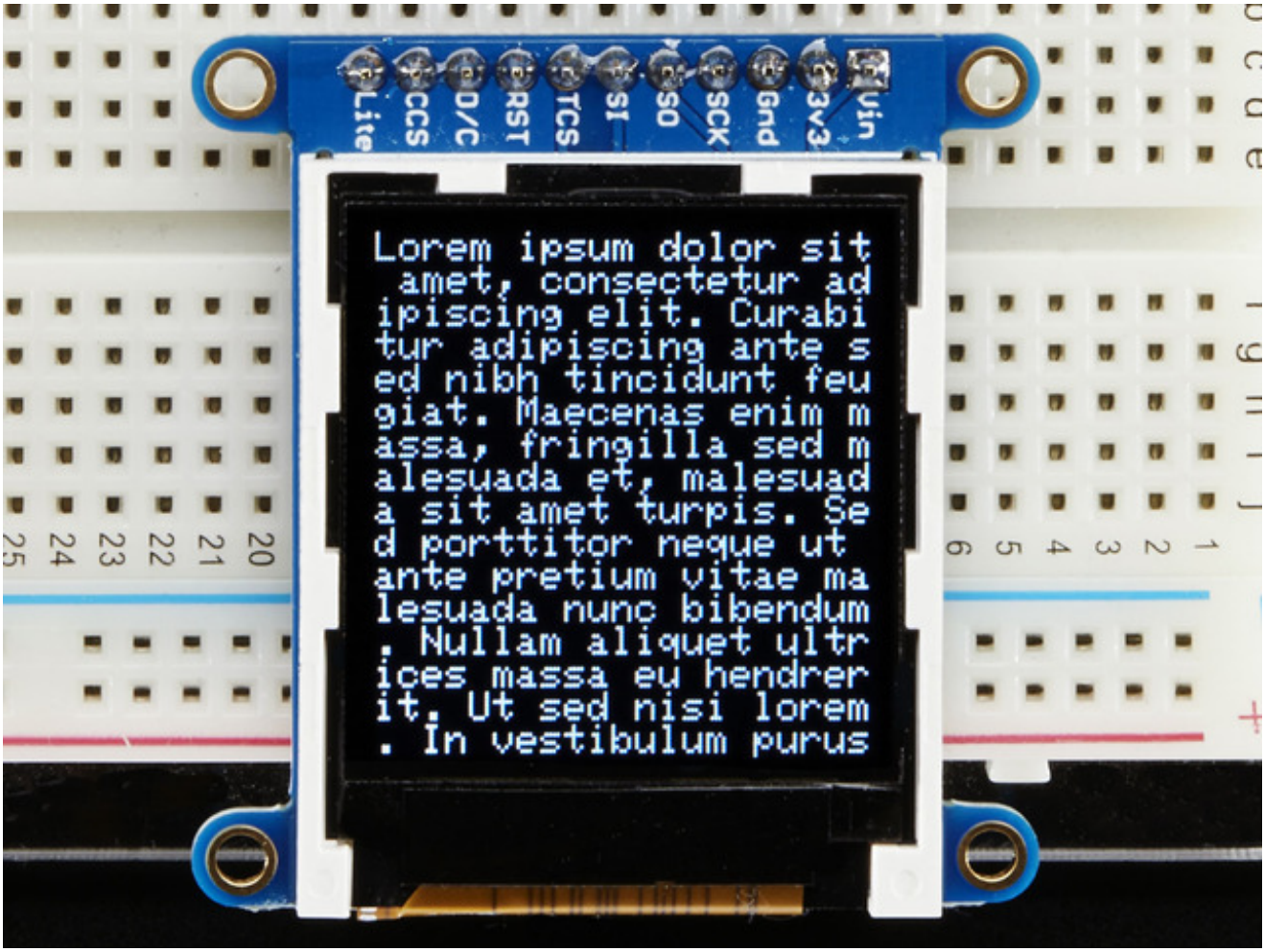
Guide Contents 2
Overview 3
Pinouts 7
Assembly 9
- Prepare the header strip: 9
- Add the breakout board: 10
- And Solder! 11

Wiring & Test 13
Wiring 13
Install Adafruit ST7735 TFT Library 14
Changing Pins 16
Adafruit GFX library 18
Drawing Bitmaps 20
Downloads 24
Files & Datasheets 24
Schematics & PCB Print 24

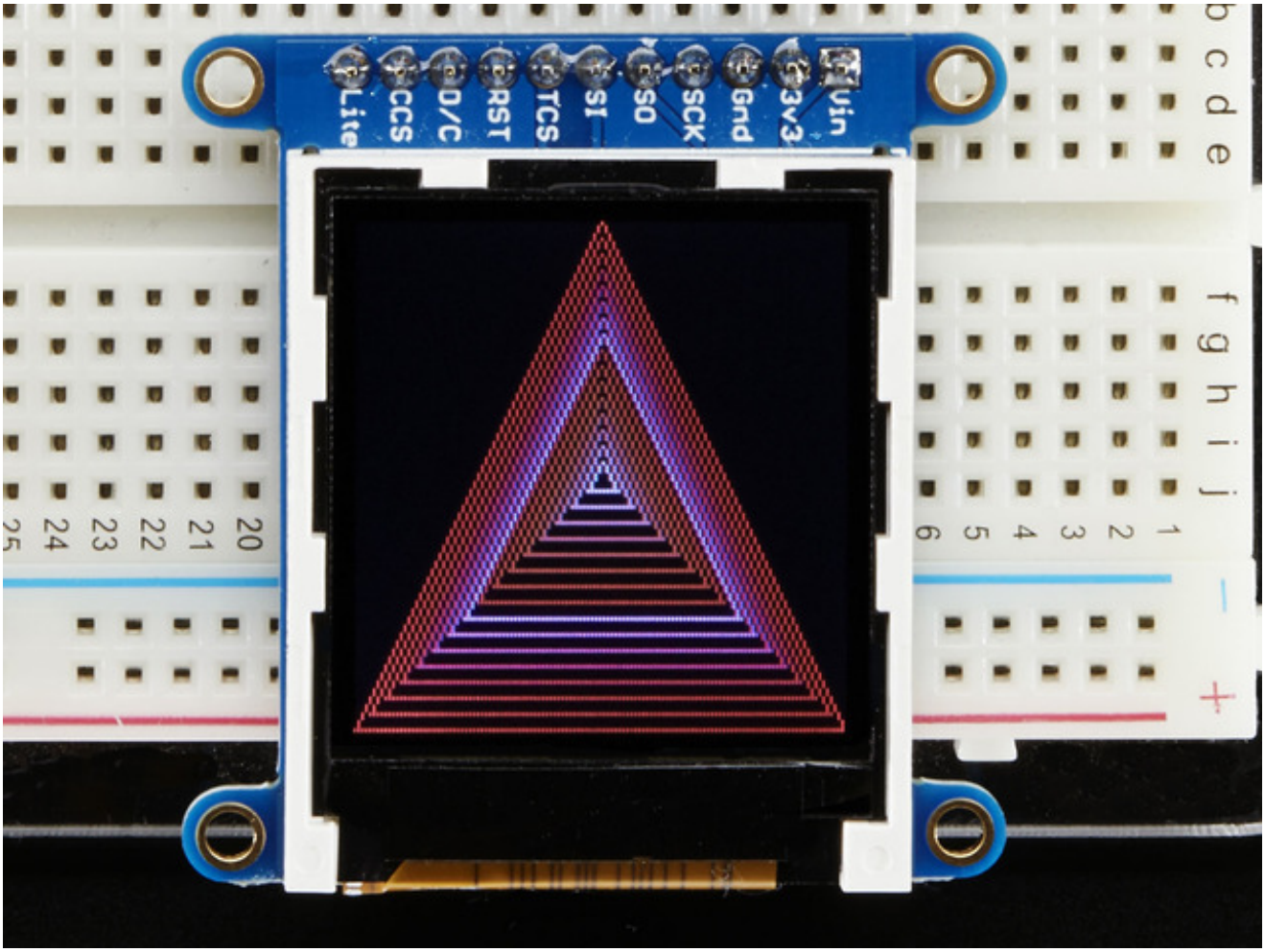
Overview



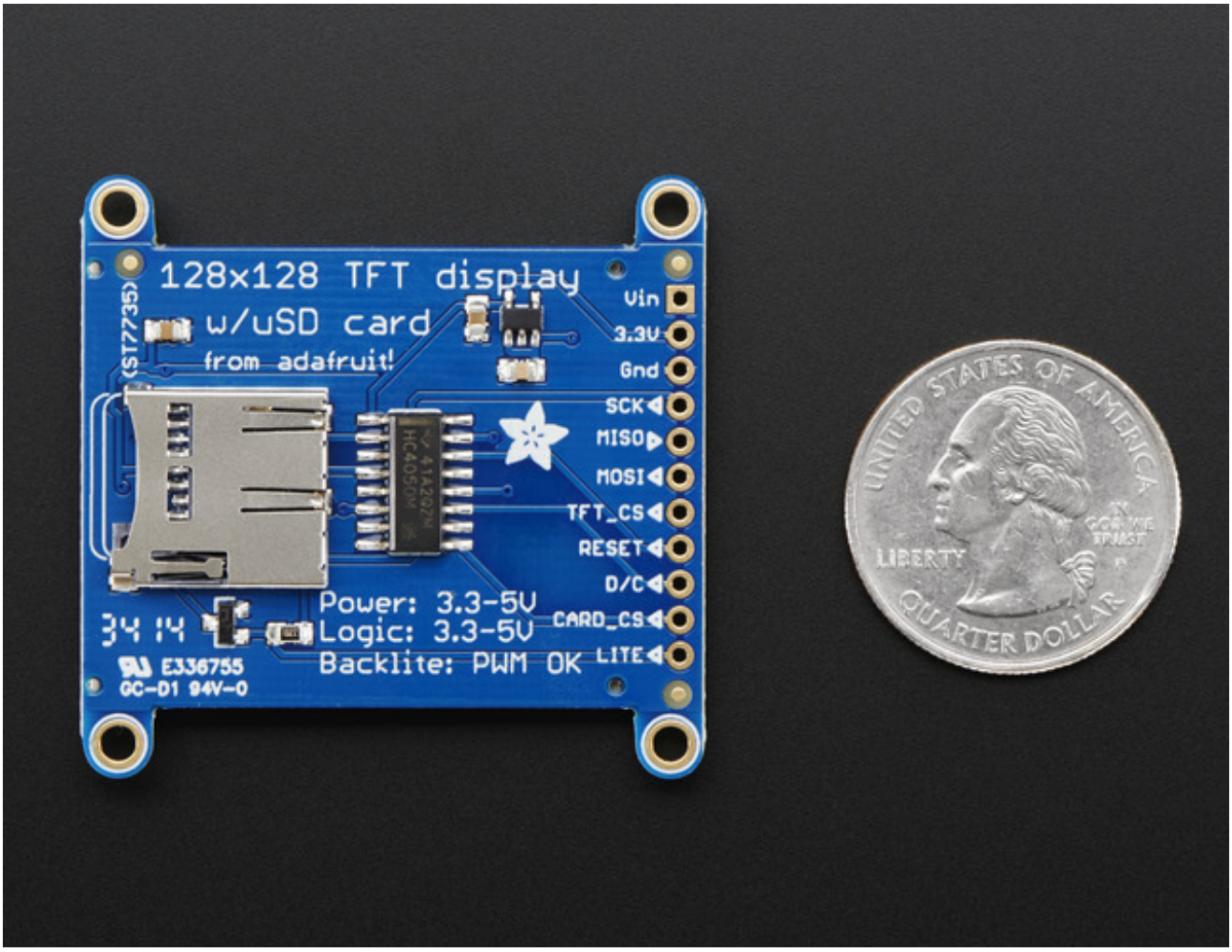
This lovely little display breakout is the best way to add a small, colorful and bright display to any project. Since the display uses 4-wire SPI to communicate and has its own pixel-addressable frame buffer, it can be used with every kind of microcontroller. Even a very small one with low memory and few pins available!



The 1.44" display has 128x128 color pixels. Unlike the low cost "Nokia 6110" and similar LCD displays, which are CSTN type and thus have poor color and slow refresh, this display is a true TFT! The TFT driver (ST7735R) can display full 16-bit color using our library code.

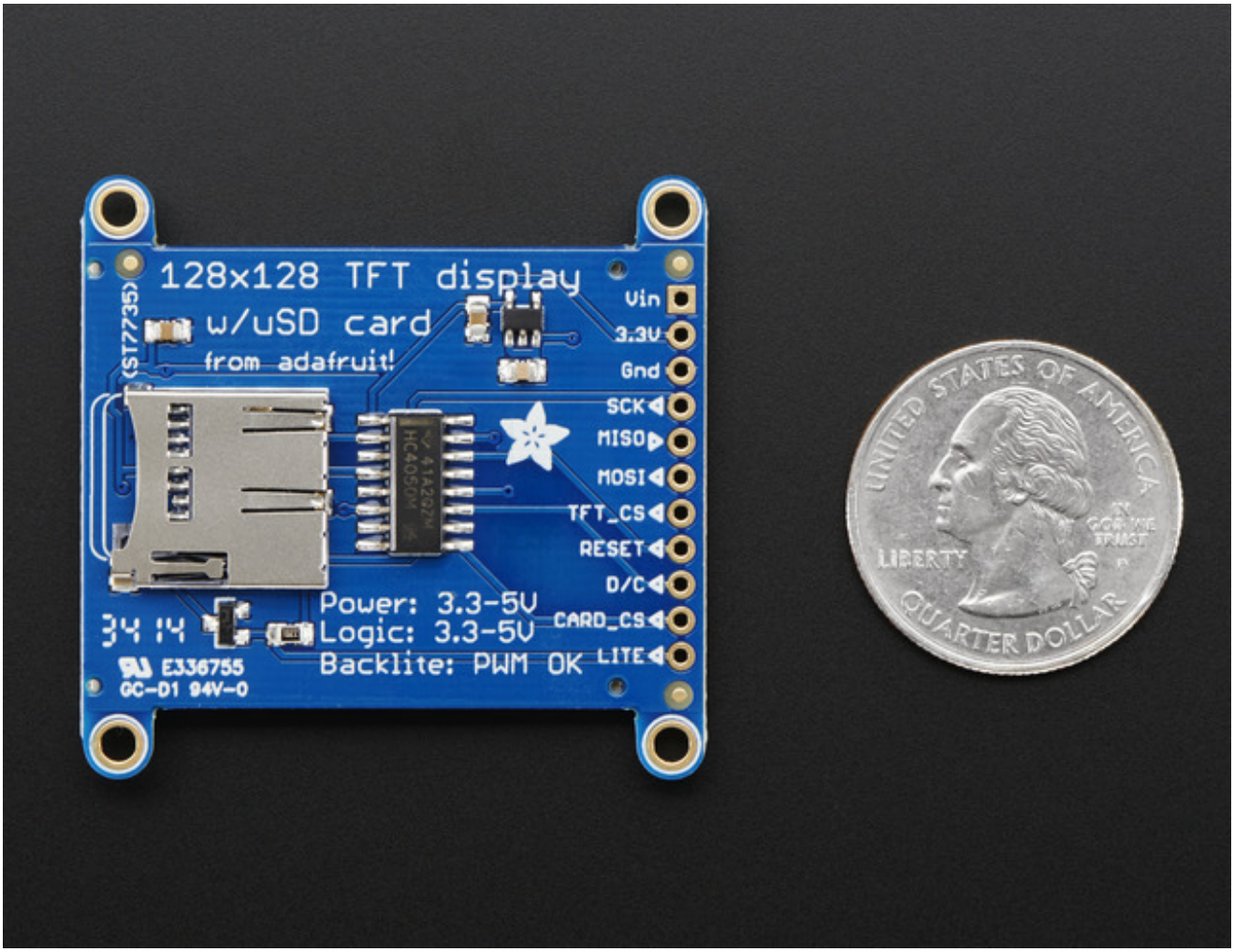


The breakout has the TFT display soldered on (it uses a delicate flex-circuit connector) as well as a ultra-low-dropout 3.3V regulator and a 3/5V level shifter so you can use it with 3.3V or 5V power and logic. We also had a little space so we placed a microSD card holder so you can easily load full color bitmaps from a FAT16/FAT32 formatted microSD card. The microSD card is not included, [but you can pick one up here](http://adafru.it/102).



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, text and bitmaps as well as example code and a wiring tutorial (http://adafru.it/ckK). The code is written for Arduino but can be easily ported to your favorite microcontroller!

Pinouts



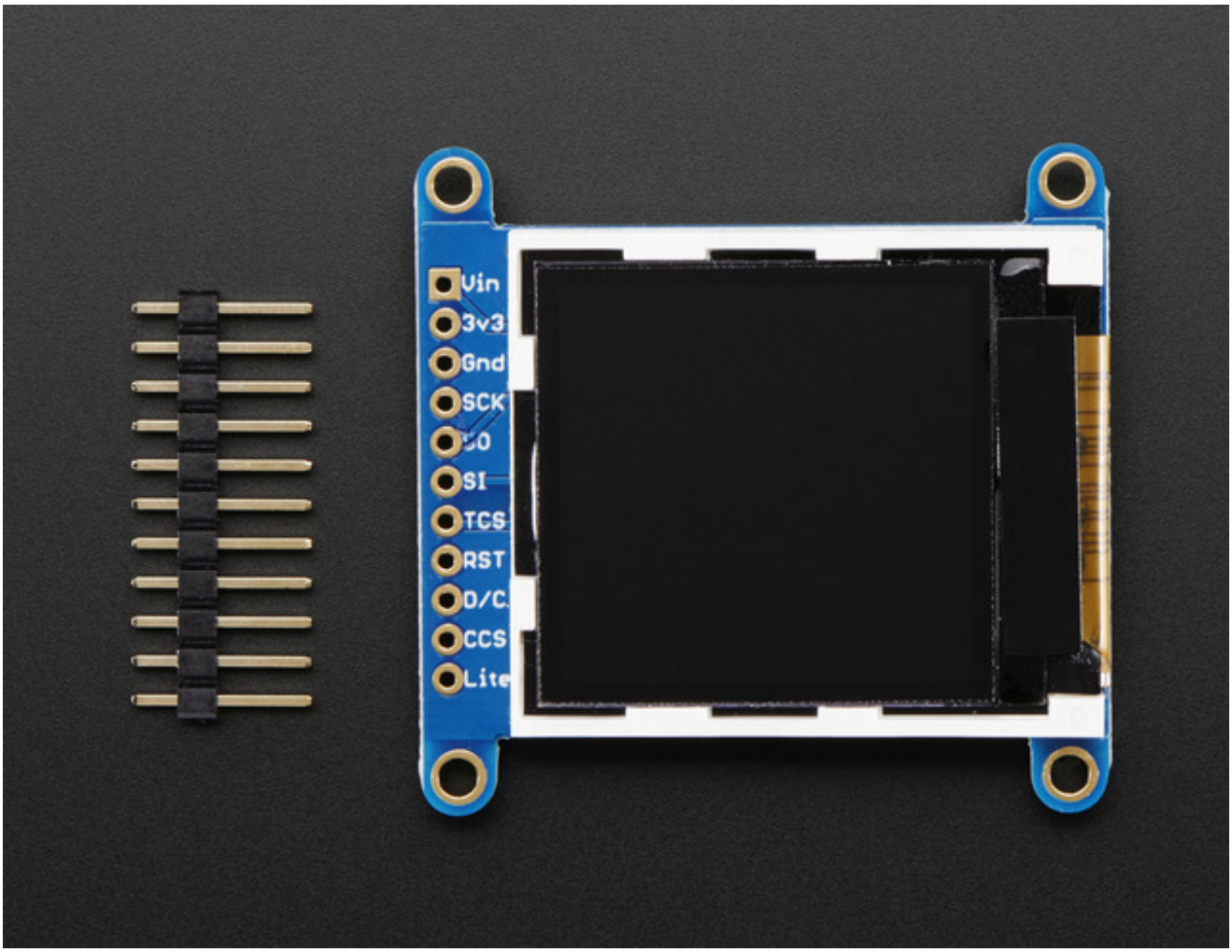
This color display uses SPI to receive image data. That means you need at least 4 pins - clock, data in, tft cs and d/c. If you'd like to have SD card usage too, add another 2 pins - data out and card cs. However, there's a couple other pins you may want to use, lets go thru them all!

- **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
- **GND** - this is the power and signal ground pin
- **CLK** - this is the SPI clock input pin
- **MISO** - this is the SPI Master In Slave Out pin, its used for the SD card. It isn't used for 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

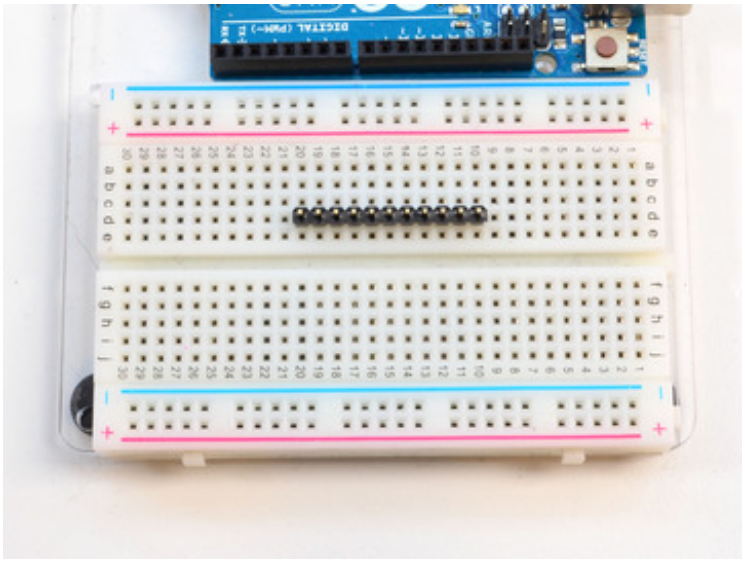
microcontroller to the SD card and/or TFT

- **TFT_CS** - this is the TFT SPI chip select pin
- **RST** - this is the TFT reset pin. Connect to ground to reset the TFT! Its best to have this pin controlled by the library so the display is reset cleanly, but you can also connect it to the Arduino Reset pin, which works for most cases.
- **D/C** - this is the TFT SPI data or command selector pin
- **Card CS / CCS - this is the SD card chip select, used if you want to read from the SD card.**
- **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

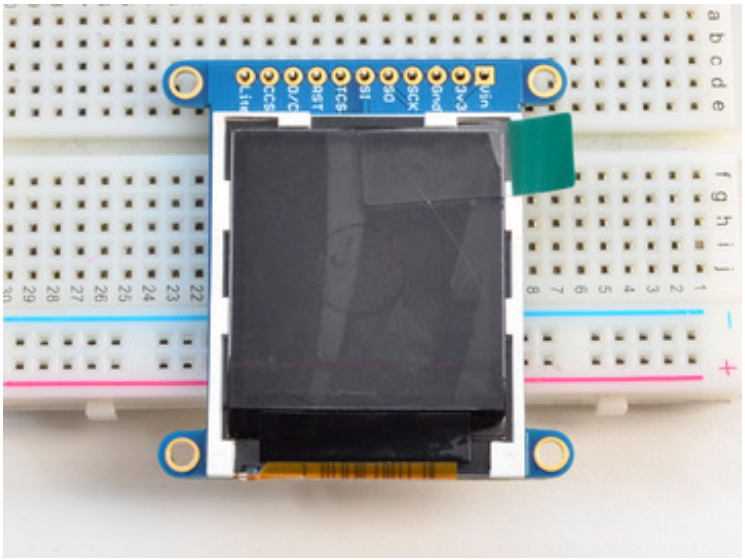
Assembly



Prepare the header strip:

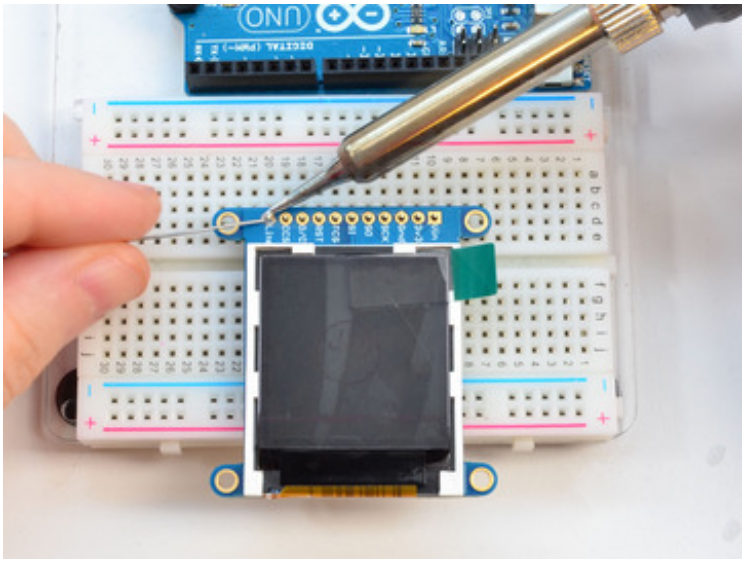
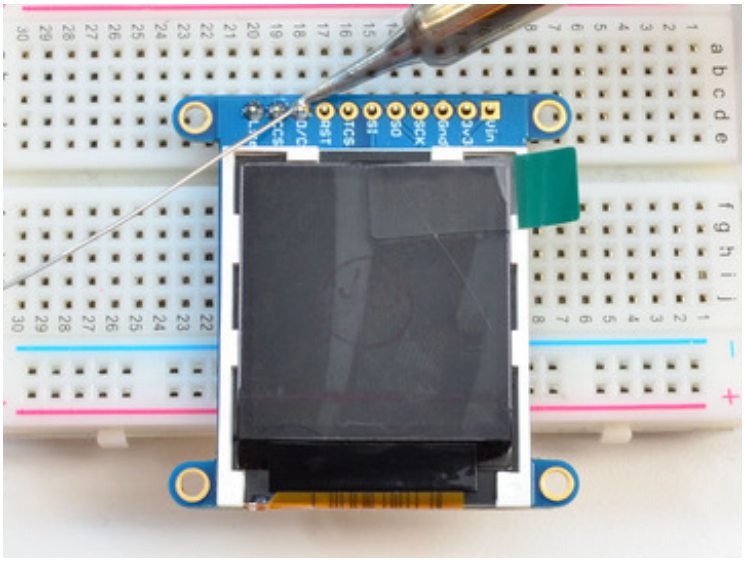
- 

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:

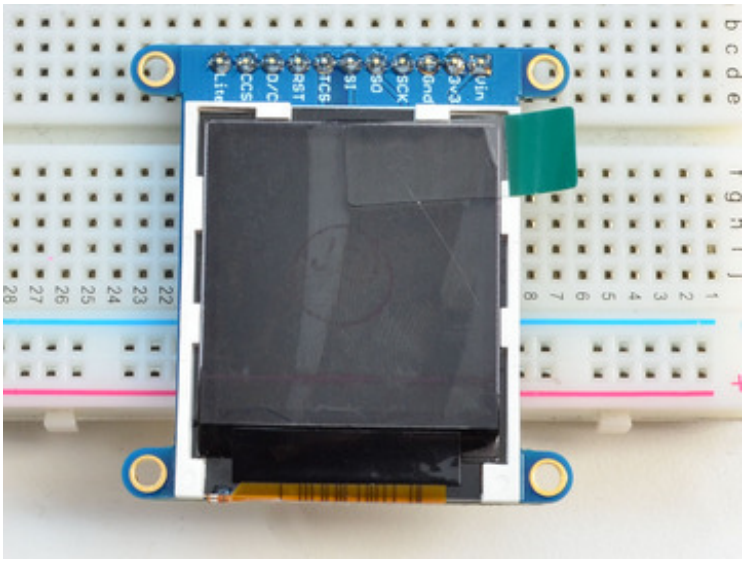
Place the breakout board over the pins so that the short pins poke through the breakout pads

- 
- 

And Solder!

Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our Guide to Excellent Soldering (http://adafru.it/aTk).

- 

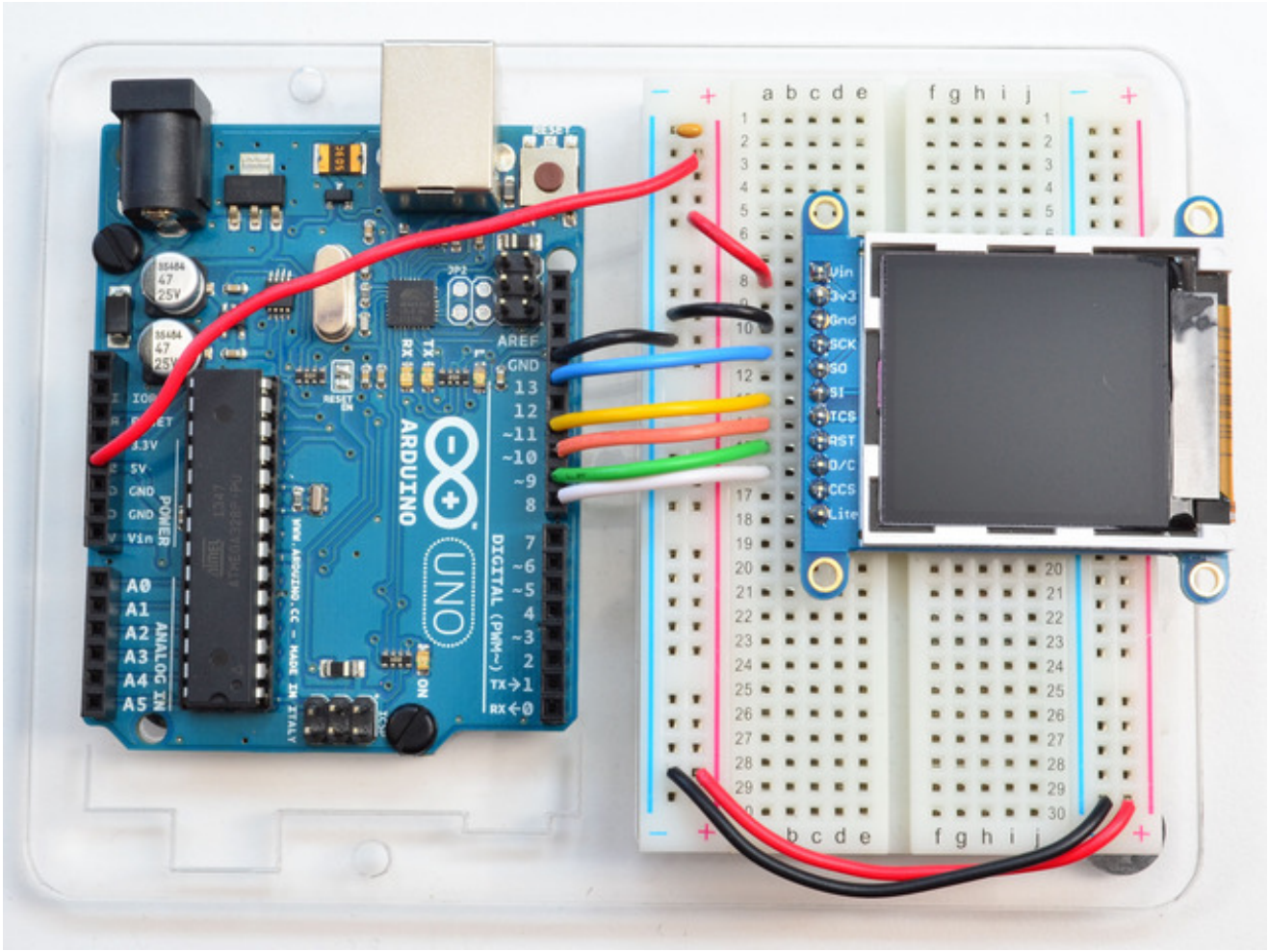
You're done! Check your solder joints visually and continue onto the next steps

Wiring & Test

Wiring

Wiring up the display in SPI mode is pretty easy as there's not that many pins! We'll be using hardware SPI, but you can also use software SPI (any pins) later. Start by connecting the power pins

- **3-5V Vin** connects to the Arduino **5V** pin
- **GND** connects to Arduino ground
- **CLK** connects to SPI clock. On Arduino Uno/Duemilanove/328-based, thats **Digital 13**. On Mega's, its **Digital 52** and on Leonardo/Due its **ICSP-3** (See SPI Connections for more details (http://adafru.it/d5h))
- **MOSI** connects to SPI MOSI. On Arduino Uno/Duemilanove/328-based, thats **Digital 11**. On Mega's, its **Digital 51** and on Leonardo/Due its **ICSP-4** (See SPI Connections for more details (http://adafru.it/d5h))
- **CS** connects to our SPI Chip Select pin. We'll be using **Digital 10** but you can later change this to any pin
- **RST** connects to our TFT reset pin. We'll be using **Digital 9** but you can later change this pin too.
- **D/C** connects to our SPI data/command select pin. We'll be using **Digital 8** but you can later change this pin too.



Install Adafruit ST7735 TFT Library

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.

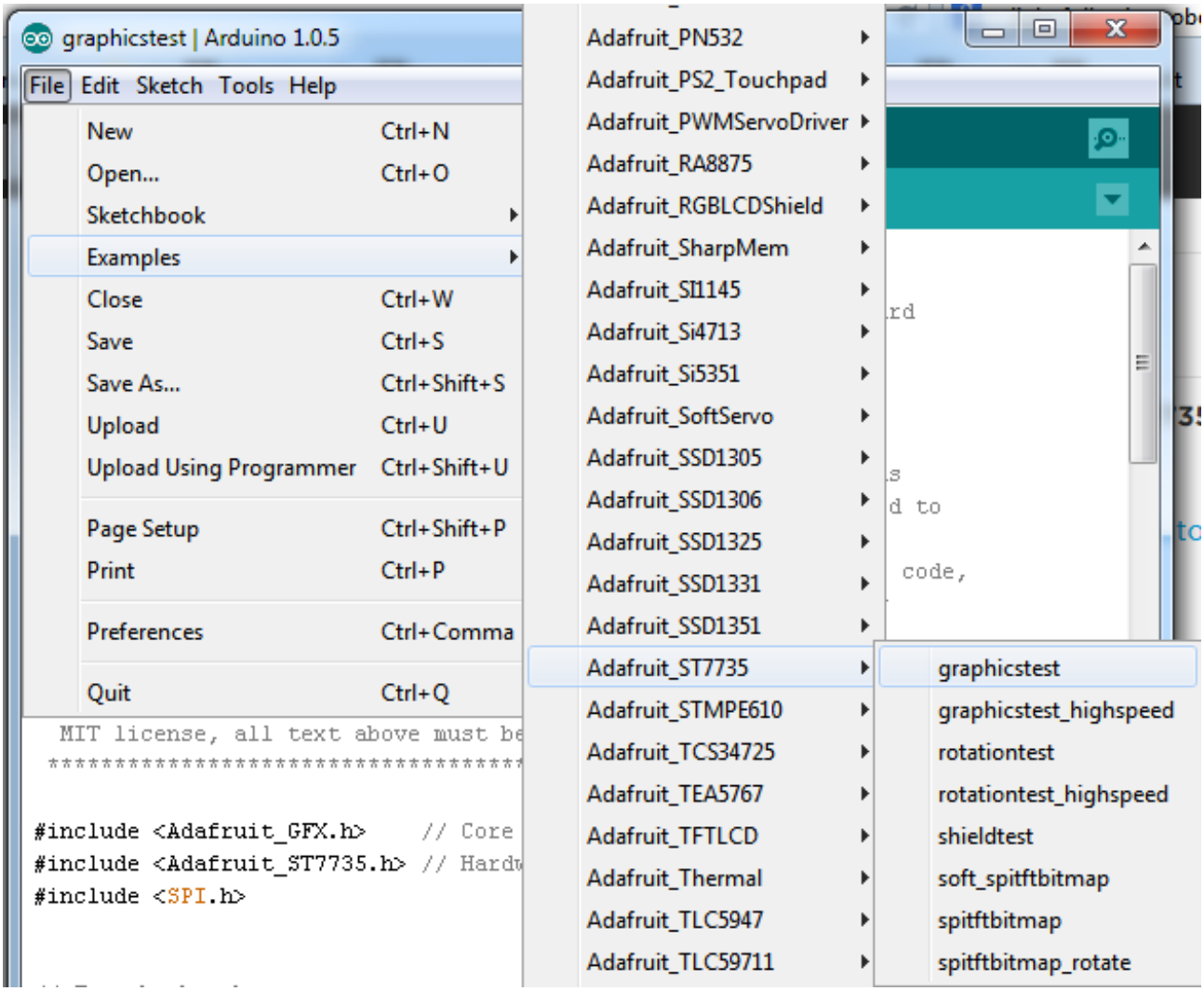
Two libraries need to be downloaded and installed: first is the Adafruit_ST7735 library (http://adafru.it/aHm) (this contains the low-level code specific to this device), and second is the Adafruit GFX Library (http://adafru.it/aJa) (which handles graphics operations common to many displays we carry). If you have **Adafruit_GFX** already, make sure its the most recent version since we've made updates for better performance

Download Adafruit ST7735 Library
http://adafru.it/dXk
Download Adafruit GFX Library
http://adafru.it/cBB

Download both ZIP files, uncompress and rename the folders to **Adafruit_ST7735**

(contains **Adafruit_ST7735.cpp** and **.h**) and **Adafruit_GFX** (contains **Adafruit_GFX.cpp** and **.h**) respectively. Then place them inside your Arduino**libraries** 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).

Restart the IDE!



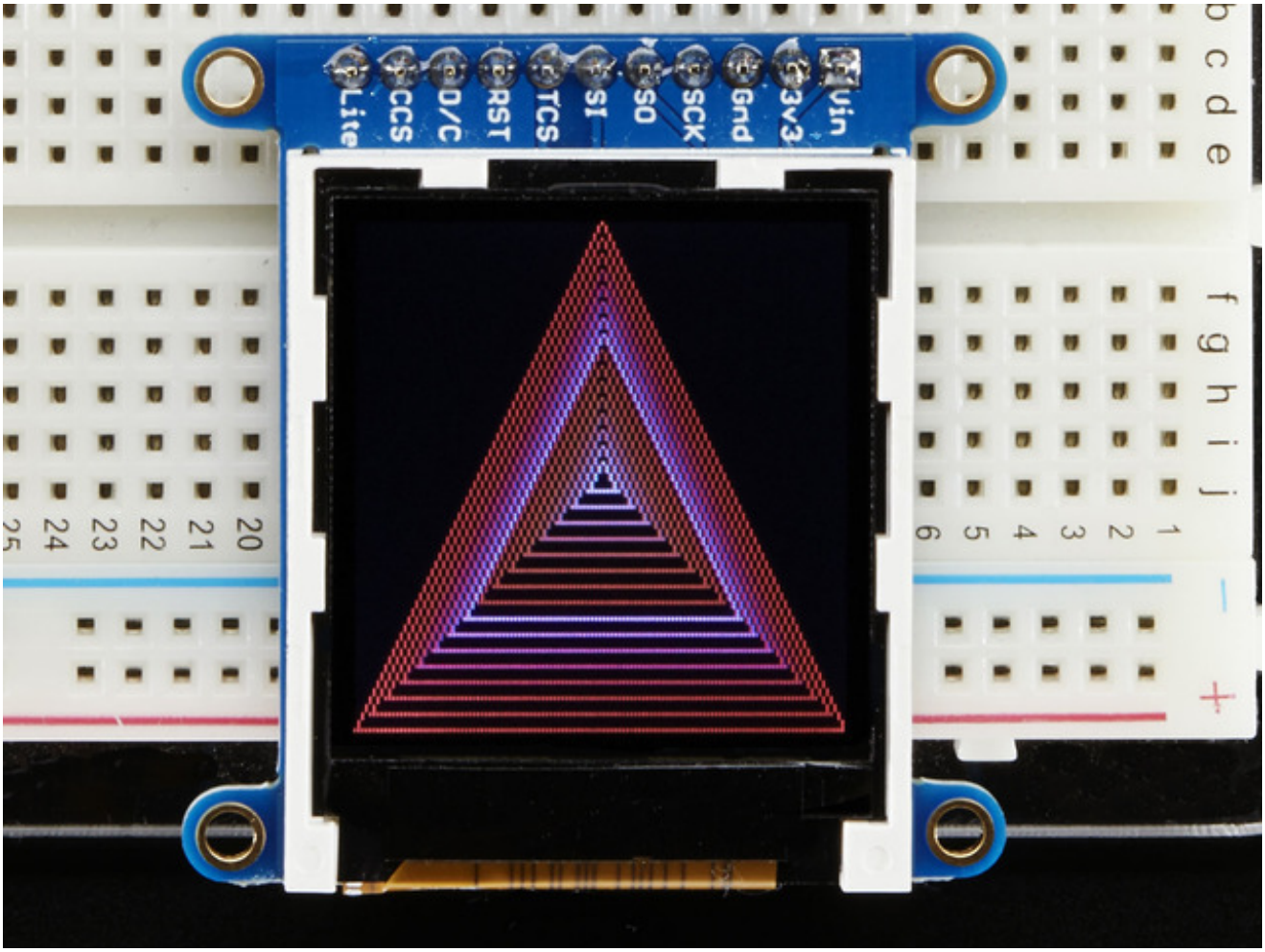
After restarting the Arduino software, you should see a new**example** folder called **Adafruit_ST7735** and inside, an example called **graphicstest**. Look for the lines as follows:

```
// Use this initializer if you're using a 1.8" TFT
tft.initR(INITR_BLACKTAB);   // initialize a ST7735S chip, black tab

// Use this initializer (uncomment) if you're using a 1.44" TFT
//tft.initR(INITR_144GREENTAB);   // initialize a ST7735S chip, black tab
```

comment out the first line, and uncomment the second, so it looks like:

```
// Use this initializer if you're using a 1.8" TFT
//tft.initR(INITR_BLACKTAB);   // initialize a ST7735S chip, black tab

// Use this initializer (uncomment) if you're using a 1.44" TFT
tft.initR(INITR_144GREENTAB);   // initialize a ST7735S chip, black tab
```

Now upload the 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.



Changing Pins

Now that you have it working, there's a few things you can do to change around the pins.

If you're using Hardware SPI, the CLOCK and MOSI pins are 'fixed' and cant be changed. But you can change to software SPI, which is a bit slower, and that lets you pick any pins you like. Find these lines:

```
// Option 1 (recommended): must use the hardware SPI pins
// (for UNO thats sclk = 13 and sid = 11) and pin 10 must be
```

```
// an output. This is much faster - also required if you want
// to use the microSD card (see the image drawing example)
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS,  TFT_DC, TFT_RST);

// Option 2: use any pins but a little slower!
#define TFT_SCLK 13   // set these to be whatever pins you like!
#define TFT_MOSI 11   // set these to be whatever pins you like!
//Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_MOSI, TFT_SCLK, TFT_RST);
```

Comment out option 1, and uncomment option 2. Then you can change the**TFT_** pins to whatever pins you'd like!

You can also save a pin by setting

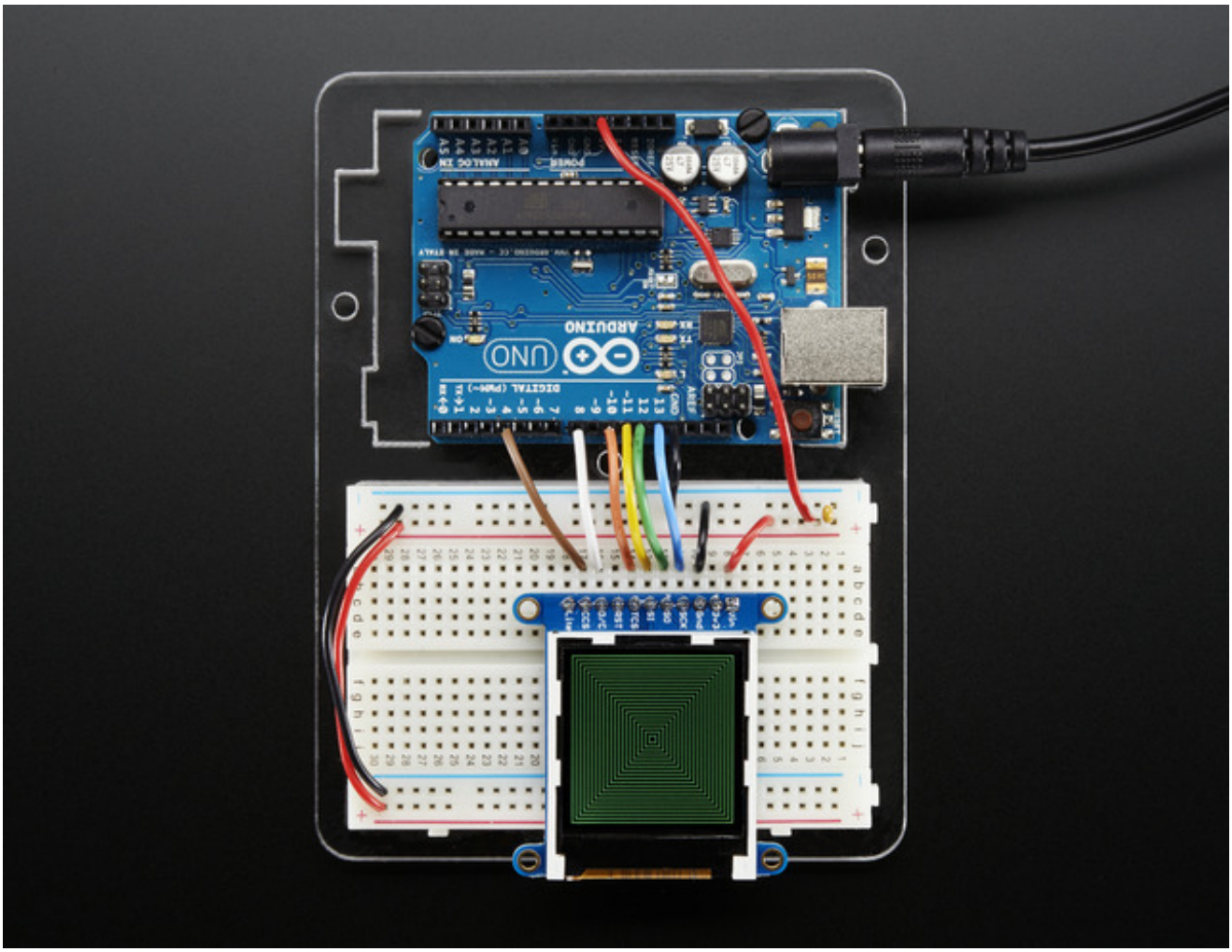
```
#define TFT_RST    9
```

to

```
#define TFT_RST   0
```

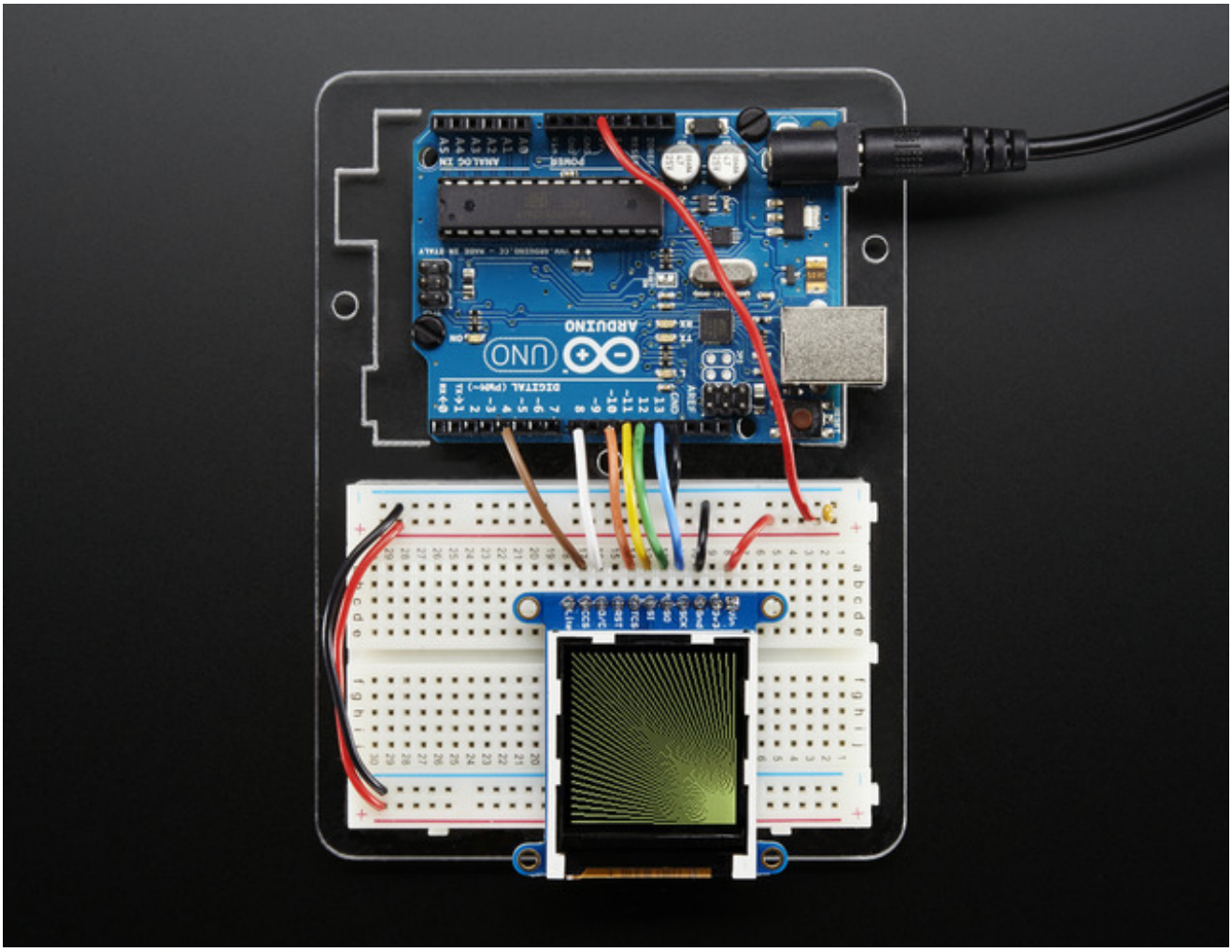
and connecting the RST line to the Arduino Reset pin. That way the Arduino will auto-reset the TFT as well.

Adafruit GFX library



The Adafruit_GFX library for Arduino provides a common syntax and set of graphics functions for all of our TFT, LCD and OLED displays. This allows Arduino sketches to easily be adapted between display types with minimal fuss…and any new features, performance improvements and bug fixes will immediately apply across our complete offering of color displays.

The GFX library is what lets you draw points, lines, rectangles, round-rects, triangles, text, etc.



Check out our detailed tutorial herehttp://learn.adafruit.com/adafruit-gfx-graphics-library (http://adafru.it/aPx) It covers the latest and greatest of the GFX library!

Drawing Bitmaps

There is a built in microSD card slot into the breakout, 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).

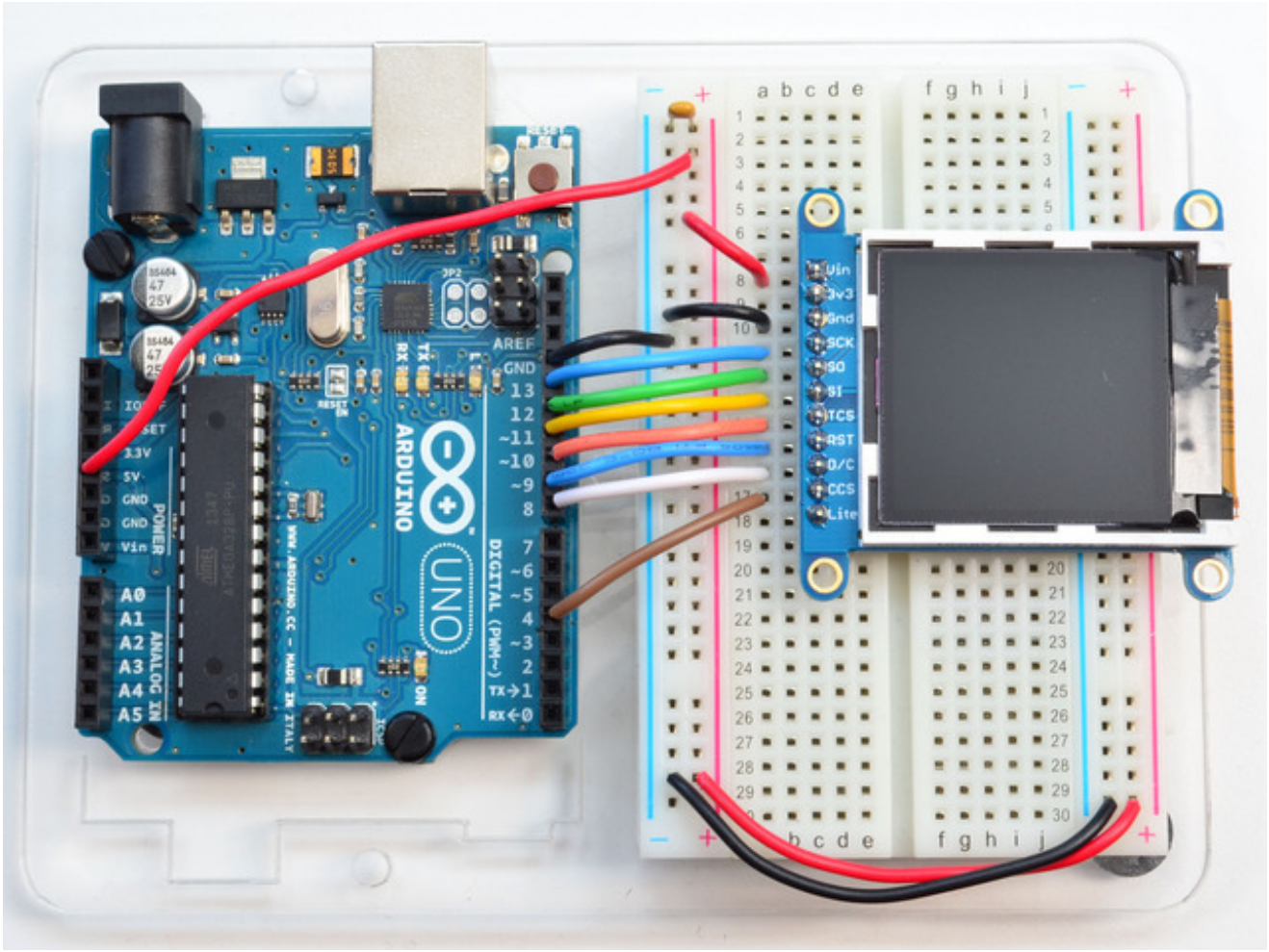
Its really easy to draw bitmaps! Lets start by downloading this image of a lily

Copy **lily128.bmp** into the base directory of a microSD card and insert it into the microSD socket in the breakout.

Two more wires are required to interface with the onboard SD card.

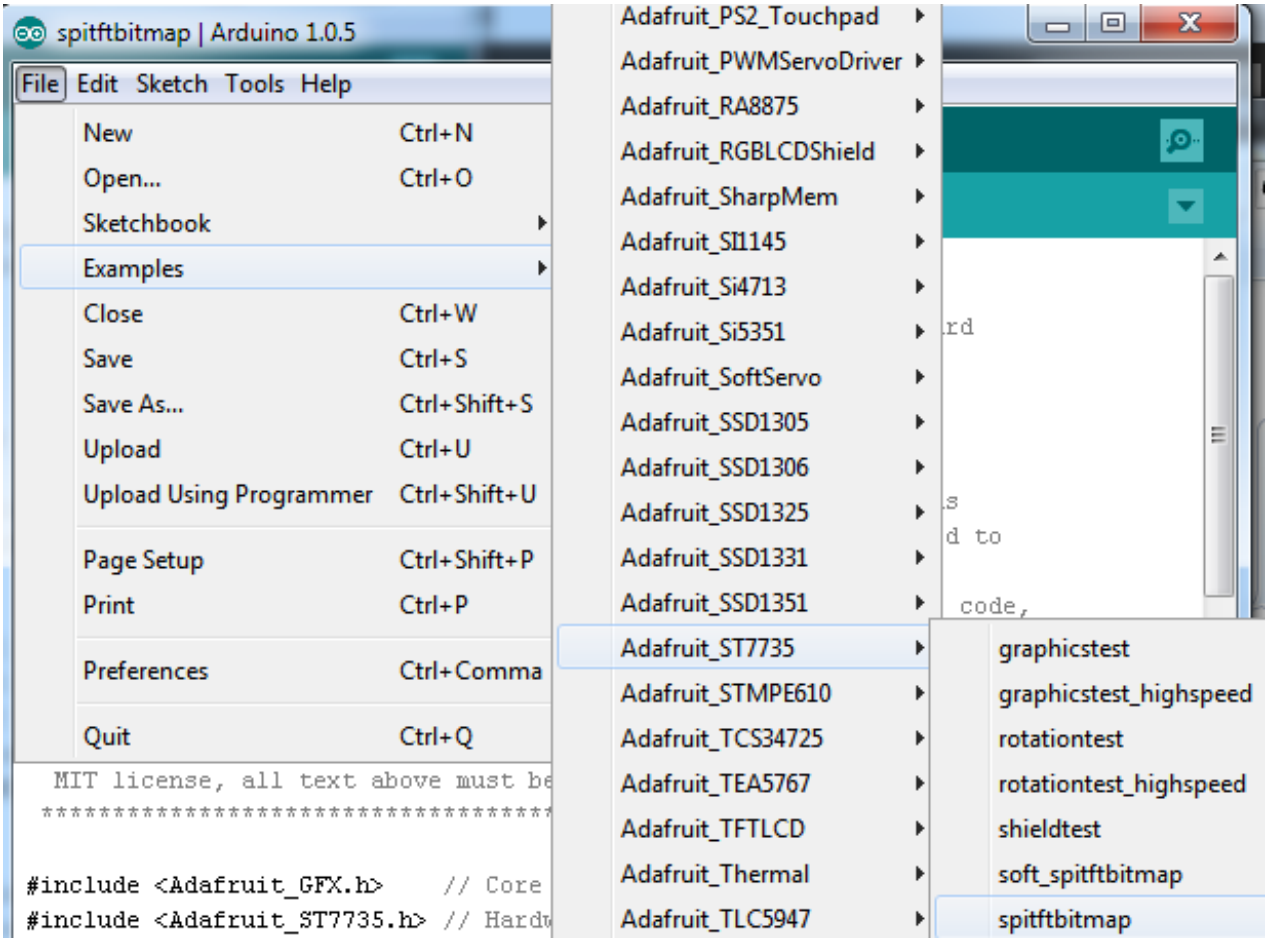
You'll need to connect up the MISO pin to the SPI MISO line on your microcontroller. On Arduino Uno/Duemilanove/328-based, thats **Digital 12**. On Mega's, its **Digital 50** and on Leonardo/Due its **ICSP-1** (See SPI Connections for more details (http://adafru.it/d5h))

Also, **CCS** pin to **Digital 4** on your Arduino as well. You can change this pin later, but stick with this for now.



You may want to try the **SD library** examples before continuing, especially one that lists all the files on the SD card

Open the **file->examples->Adafruit_ST7735->spitftbitmap** example



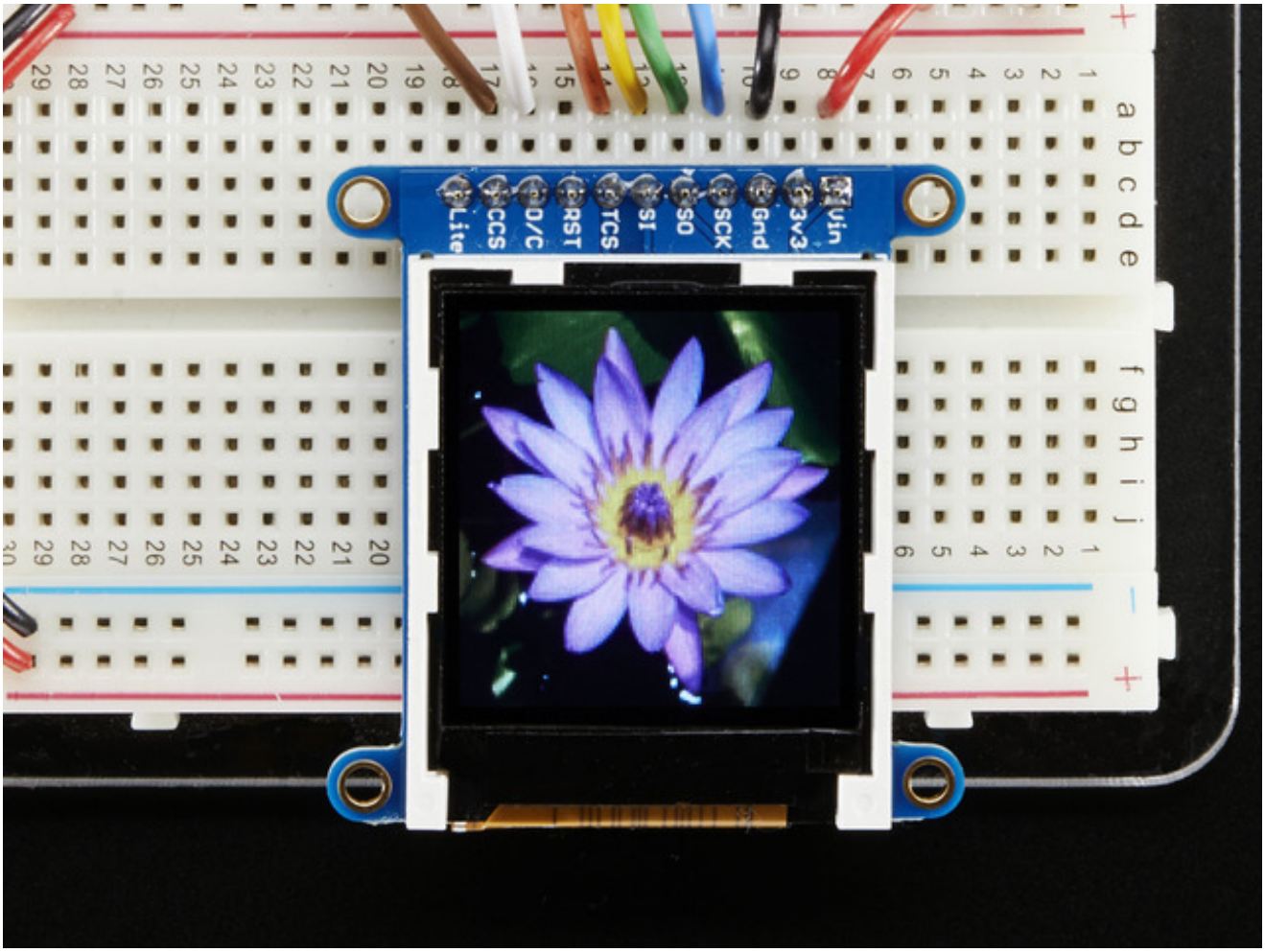
Edit the **setup()** intialization to:

```
// Use this initializer if you're using a 1.8" TFT
//tft.initR(INITR_BLACKTAB);   // initialize a 1.8" TFT

// Use this initializer (uncomment) if you're using a 1.44" TFT
tft.initR(INITR_144GREENTAB);   // initialize a 1.44" TFT
```

And the drawBmp() routine to:

```
// change the name here!
bmpDraw("lily128.bmp", 0, 0);
```

Upload to the Arduino & you will see the flowers appear!



To make new bitmaps, make sure they are less than 128 by 128 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);

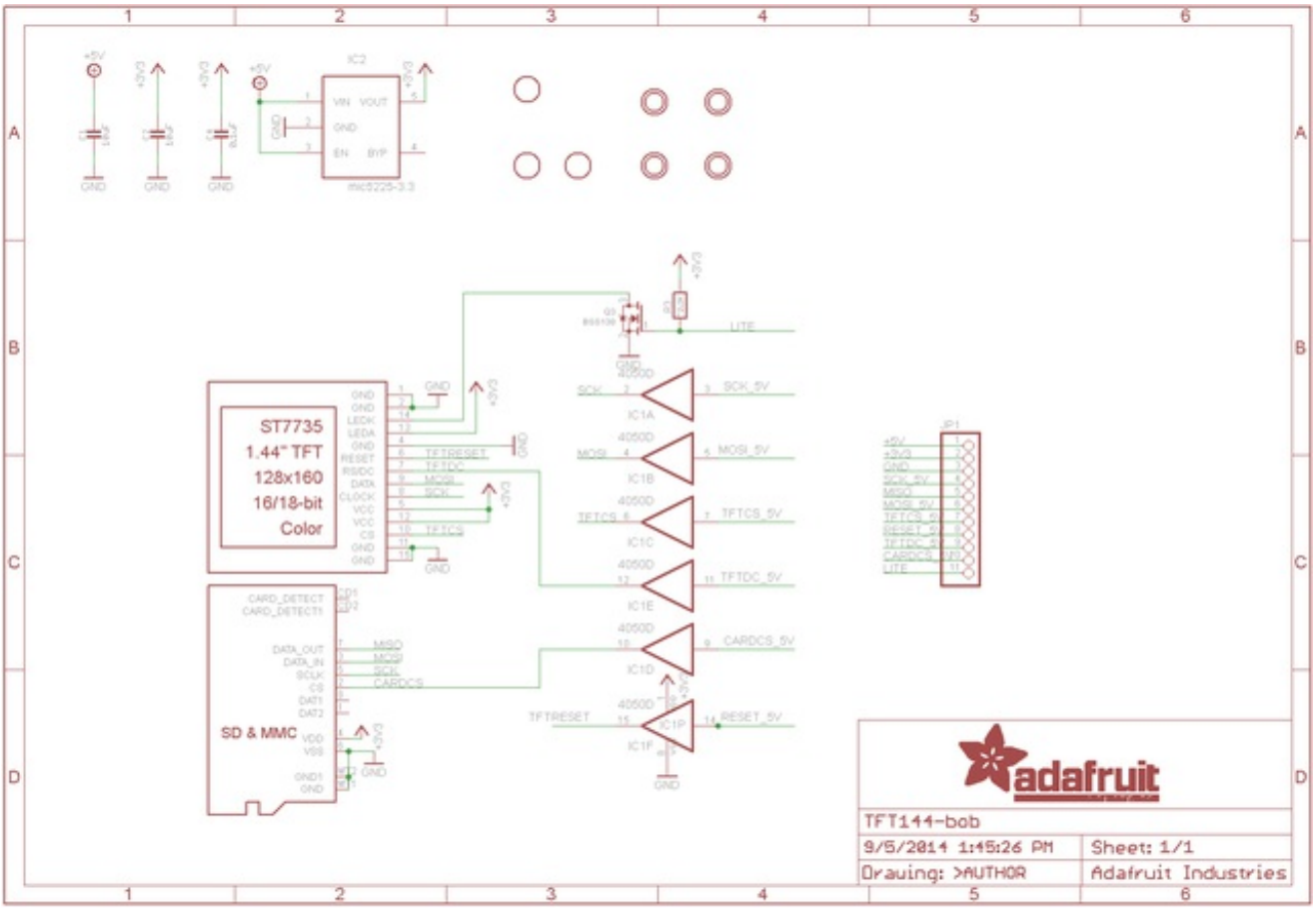
For each bitmap. They can be smaller than 128x128 and placed in any location on the screen.

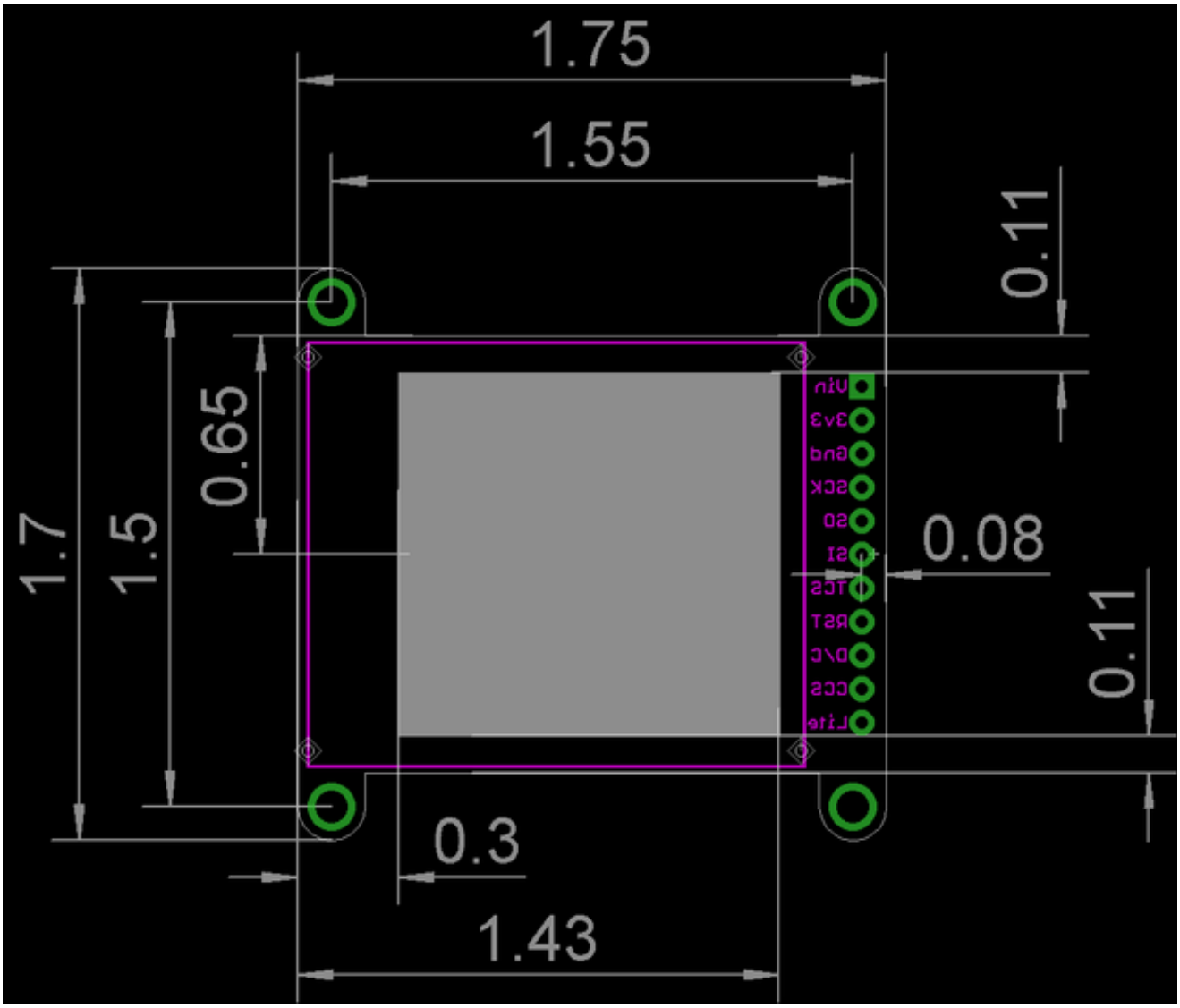
Downloads

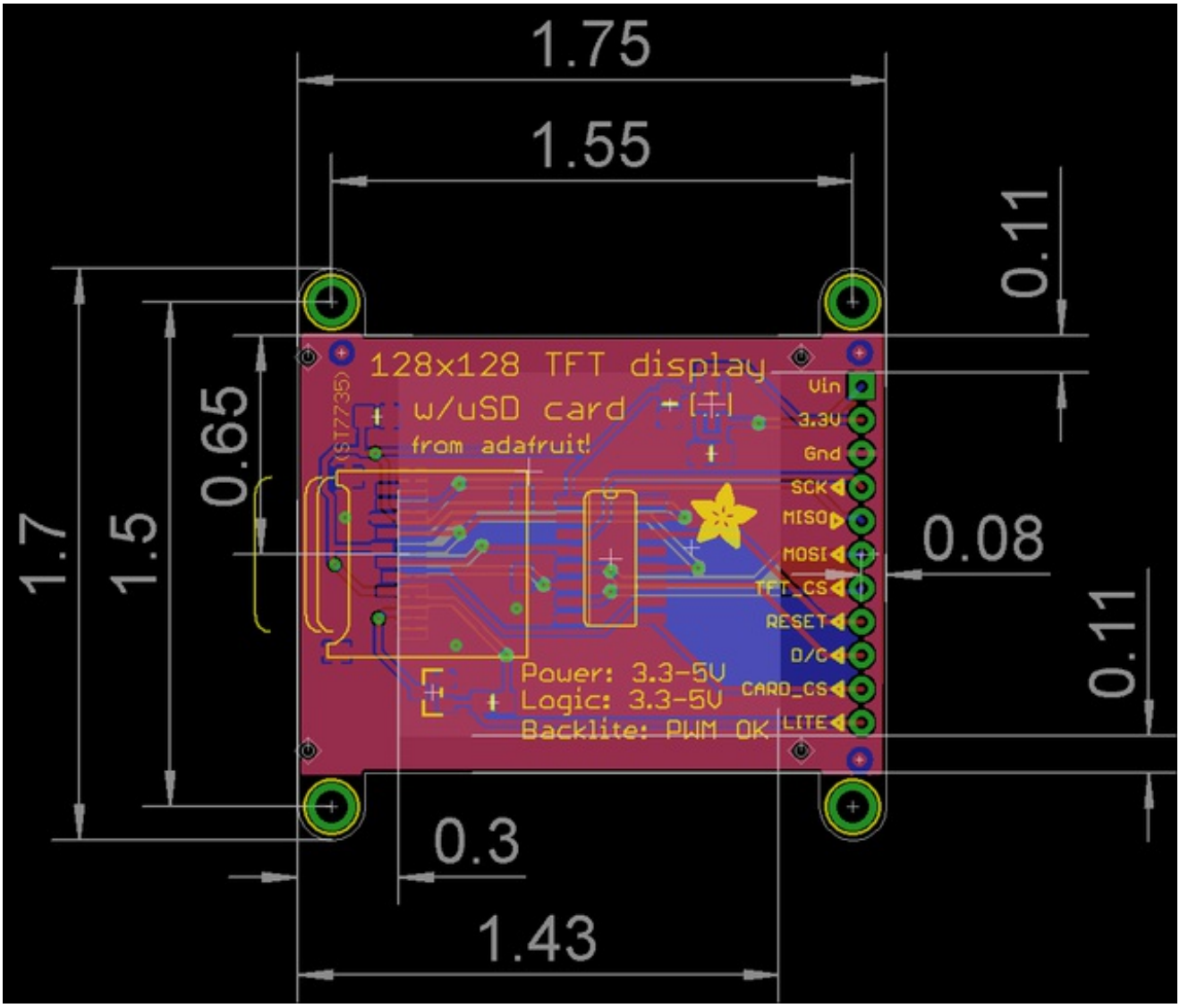
Files & Datasheets

- ST7735R display driver datasheet (http://adafru.it/aP9)
- Raw 1.44" TFT Datasheet (http://adafru.it/dYA)
- Fritzing object in Adafruit Fritzing library (http://adafru.it/aP3)
- EagleCAD PCB files in GitHub (http://adafru.it/pAu)

Schematics & PCB Print




1.75
1.55
0.11
0.65
1.7
1.5
0.08
0.11
0.3
1.43
Vin
3v3
Gnd
SCK
SO
SI
TCS
RST
D/C
CCS
Lite


1.75
1.55
0.11
0.65
1.7
1.5
0.08
0.11
0.3
1.43
128x128 TFT display
w/uSD card
from adafruit!
ST7735
Vin
3.3V
Gnd
SCK
MISO
MOSI
TFT_CS
RESET
D/C
CARD_CS
LITE
Power: 3.3-5V
Logic: 3.3-5V
Backlite: PWM OK