

□

Adafruit 2.4" TFT FeatherWing

Created by lady ada

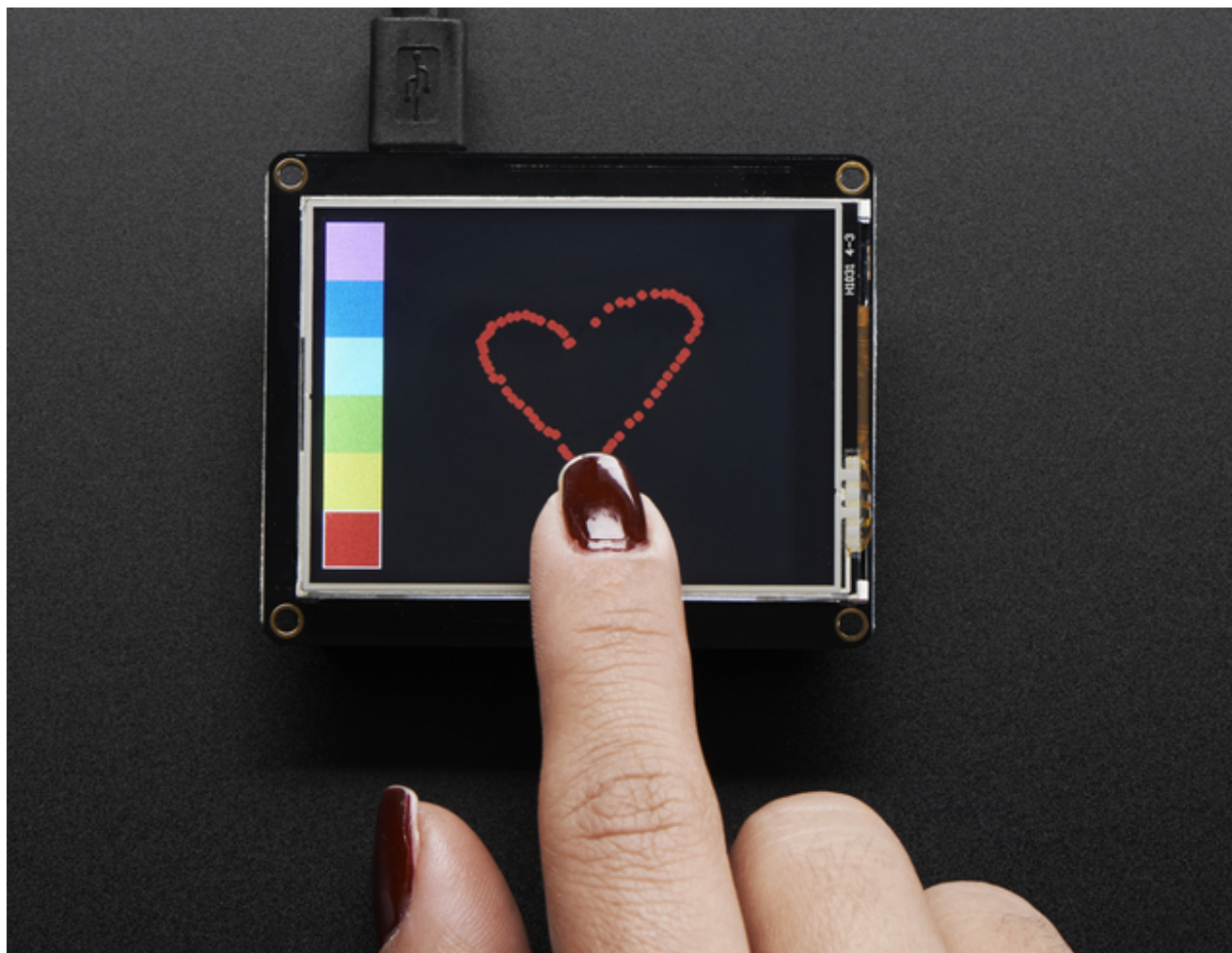


Last updated on 2016-11-09 10:44:18 PM UTC

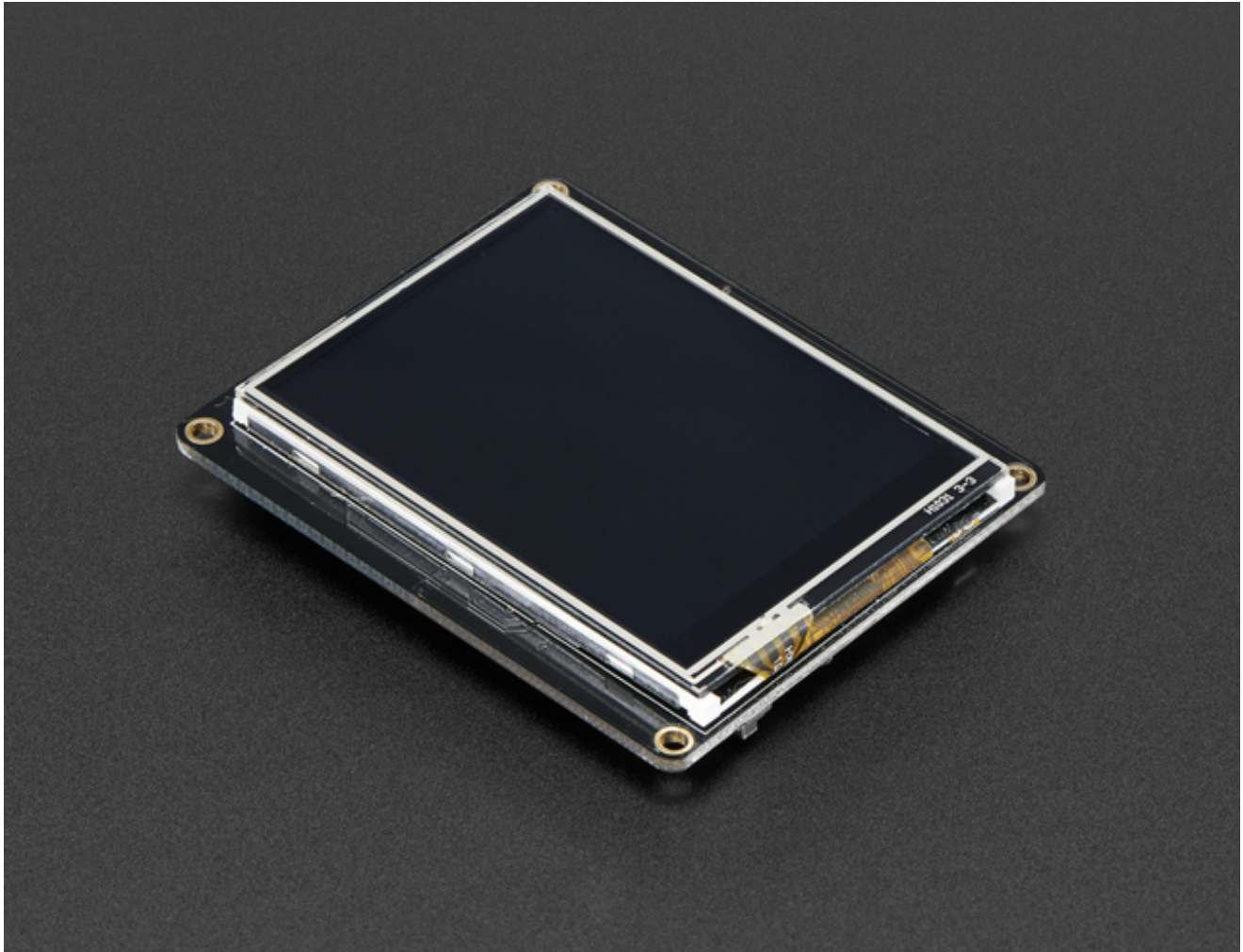
Guide Contents

Guide Contents	2
Overview	3
Pinouts	7
Power Pins	7
SPI Pins	8
TFT Control Pins	9
Touch Screen control pins	11
SD Card control pins	12
TFT Graphics Test	14
Install Libraries	14
Install Adafruit ILI9341 TFT Library	14
Basic Graphics Test	14
Adafruit GFX Library	19
Resistive Touch Screen	21
Touchscreen Paint Demo	21
Drawing Bitmaps	25
Downloads	27
Datasheets & More	27
Schematic	27
Fabrication Print	27

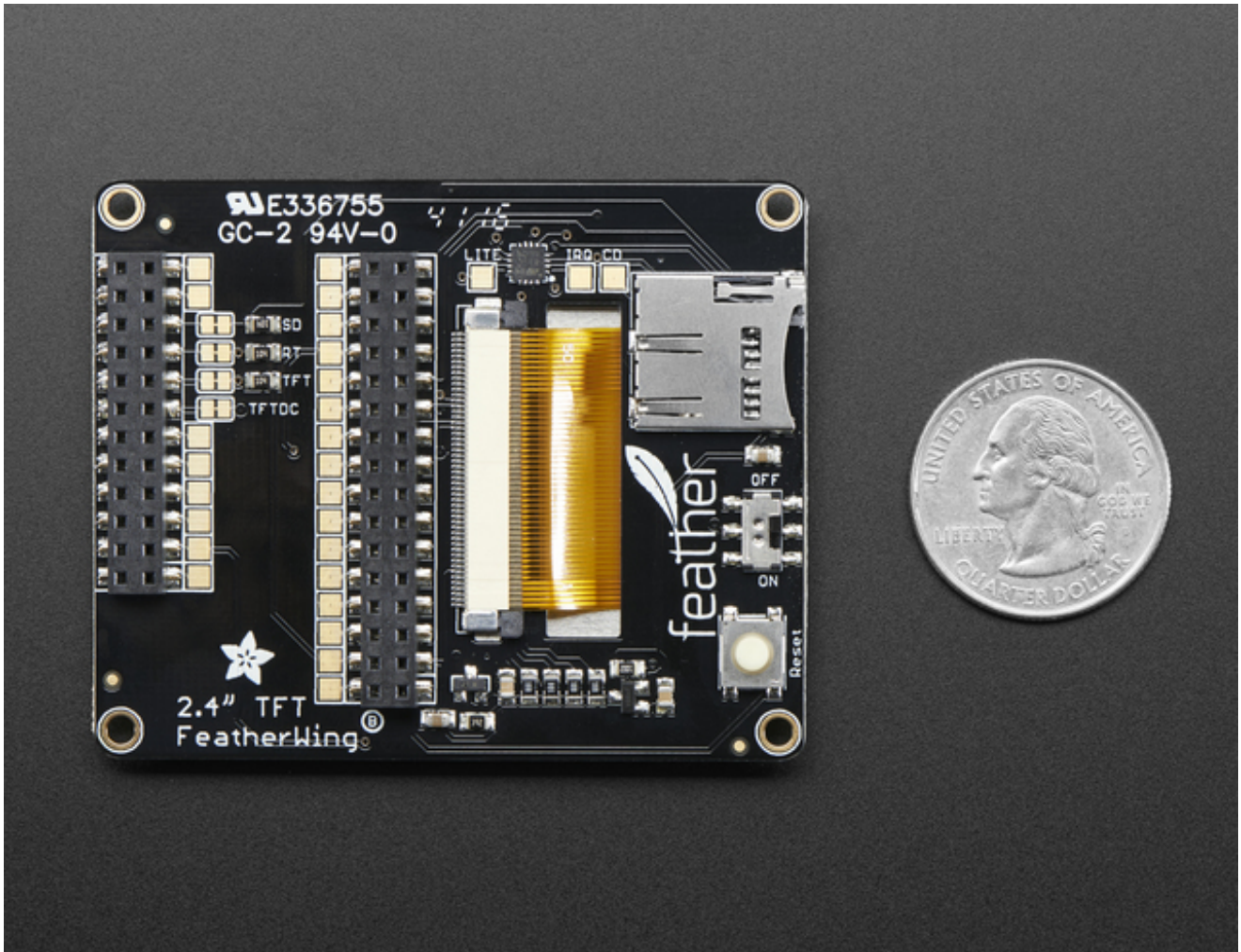
Overview



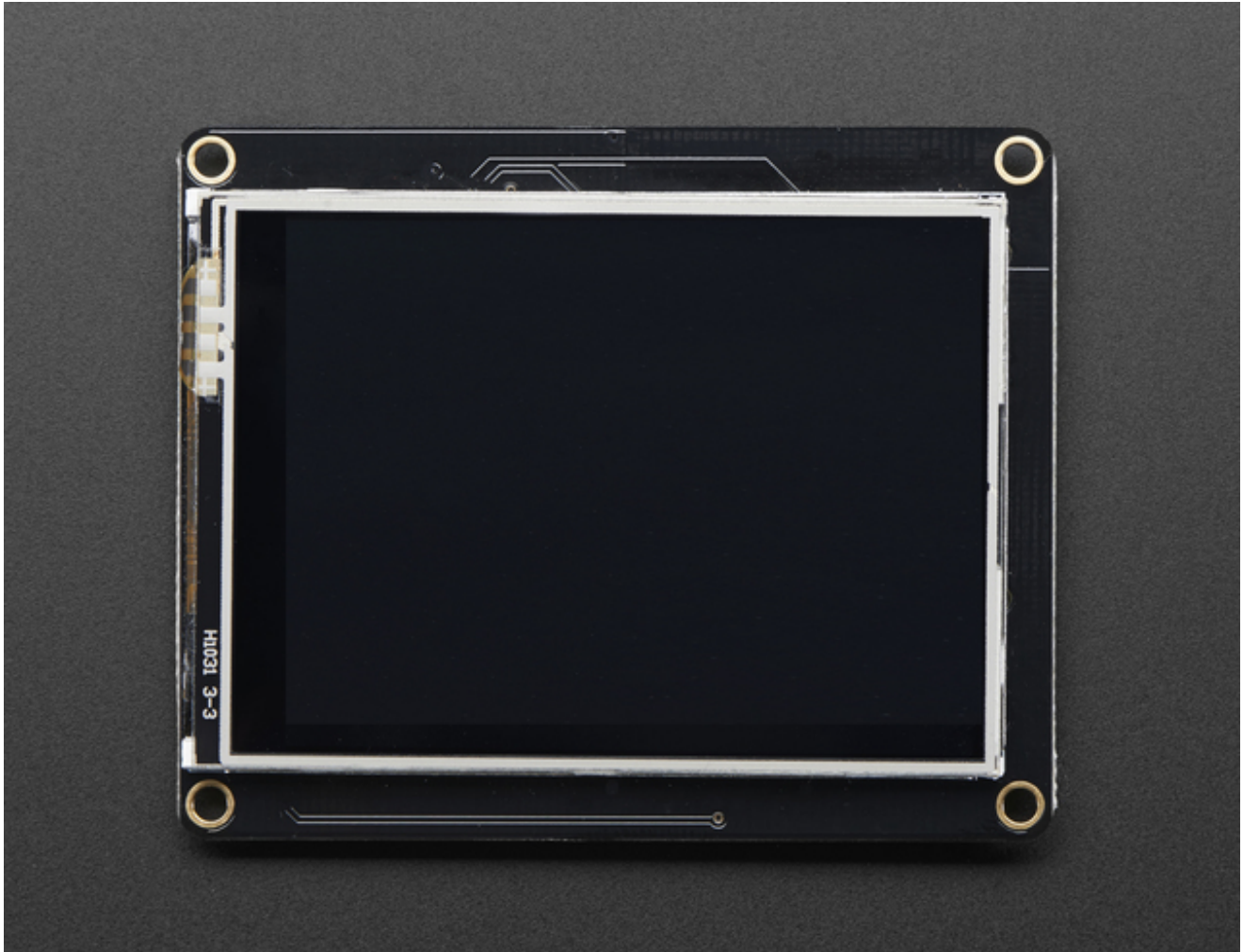
A Feather board without ambition is a Feather board without FeatherWings! Spice up your Feather project with a beautiful 2.4" touchscreen display shield with built in microSD card socket. This TFT display is 2.4" diagonal with a bright 4 white-LED backlight. You get 240x320 pixels with individual 16-bit color pixel control. It has way more resolution than a black and white 128x64 display. As a bonus, this display comes with a resistive touchscreen attached to it already, so you can detect finger presses anywhere on the screen.



This FeatherWing uses a SPI display, touchscreen and SD card socket so it works nice and fast on our **ESP8266, 32u4, M0, WICED and Teensy 3.x Feathers**. We also include an SPI resistive touchscreen controller so you only need one additional pin to add a high quality touchscreen controller. One more pin is used for an optional SD card that can be used for storing images for display.



This Wing comes fully assembled with dual sockets for your Feather to plug into. You get two sockets per pin so you can plug in wires if you want to connect to Feather pins. Alternatively, each pin has a large square pad on the PCB for direct soldering.



Four mounting holes make it easy to attach this Wing anywhere. We also include a big reset button and an on/off switch connected to the Feather Enable pin (note that the Teensy 3.x Feather does not use the Enable pin so the switch will not do anything with that type).

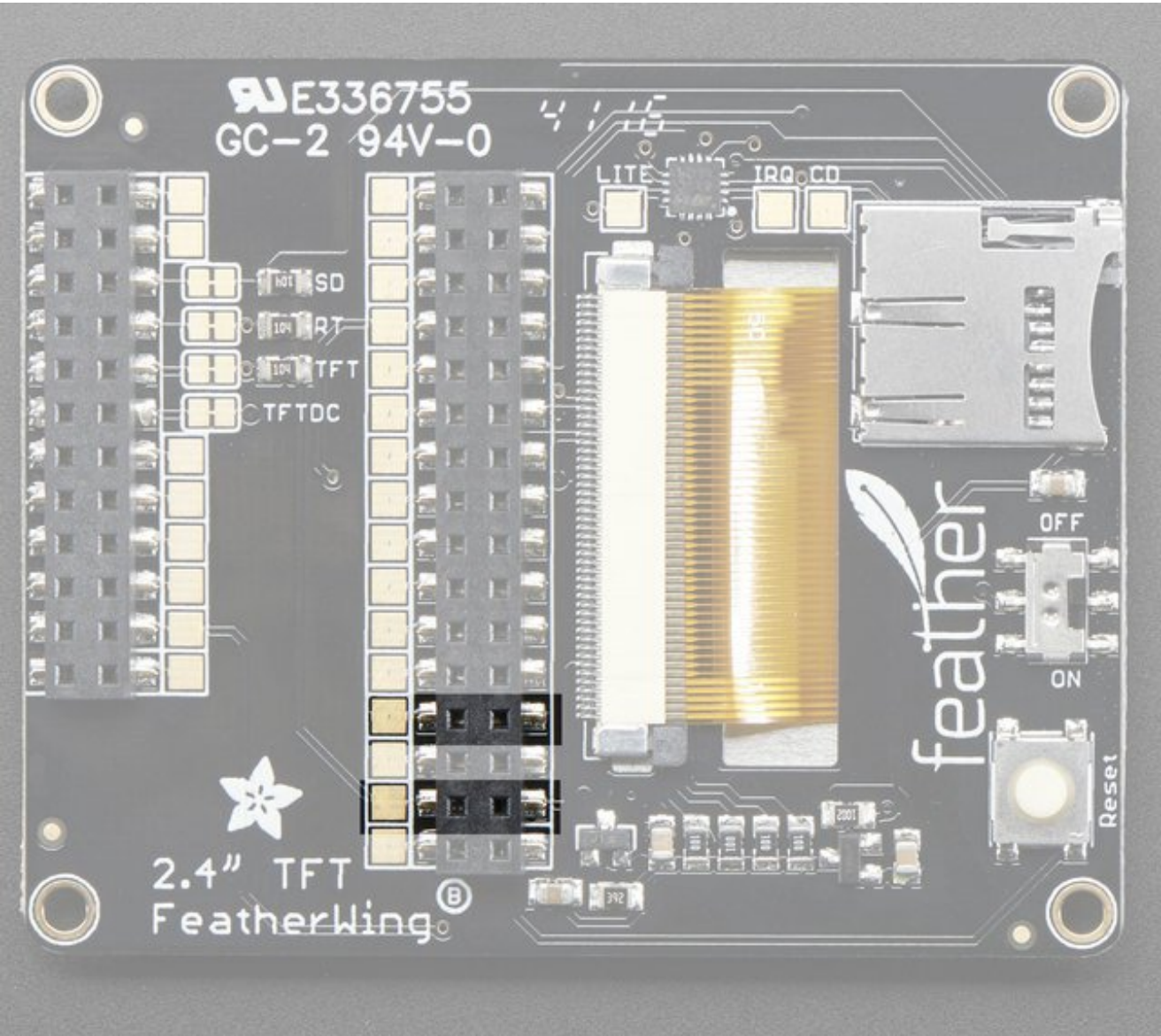


Pinouts

Unlike most FeatherWings, the TFT FeatherWing is fully assembled and has a dual socket set for your Feather to plug into.

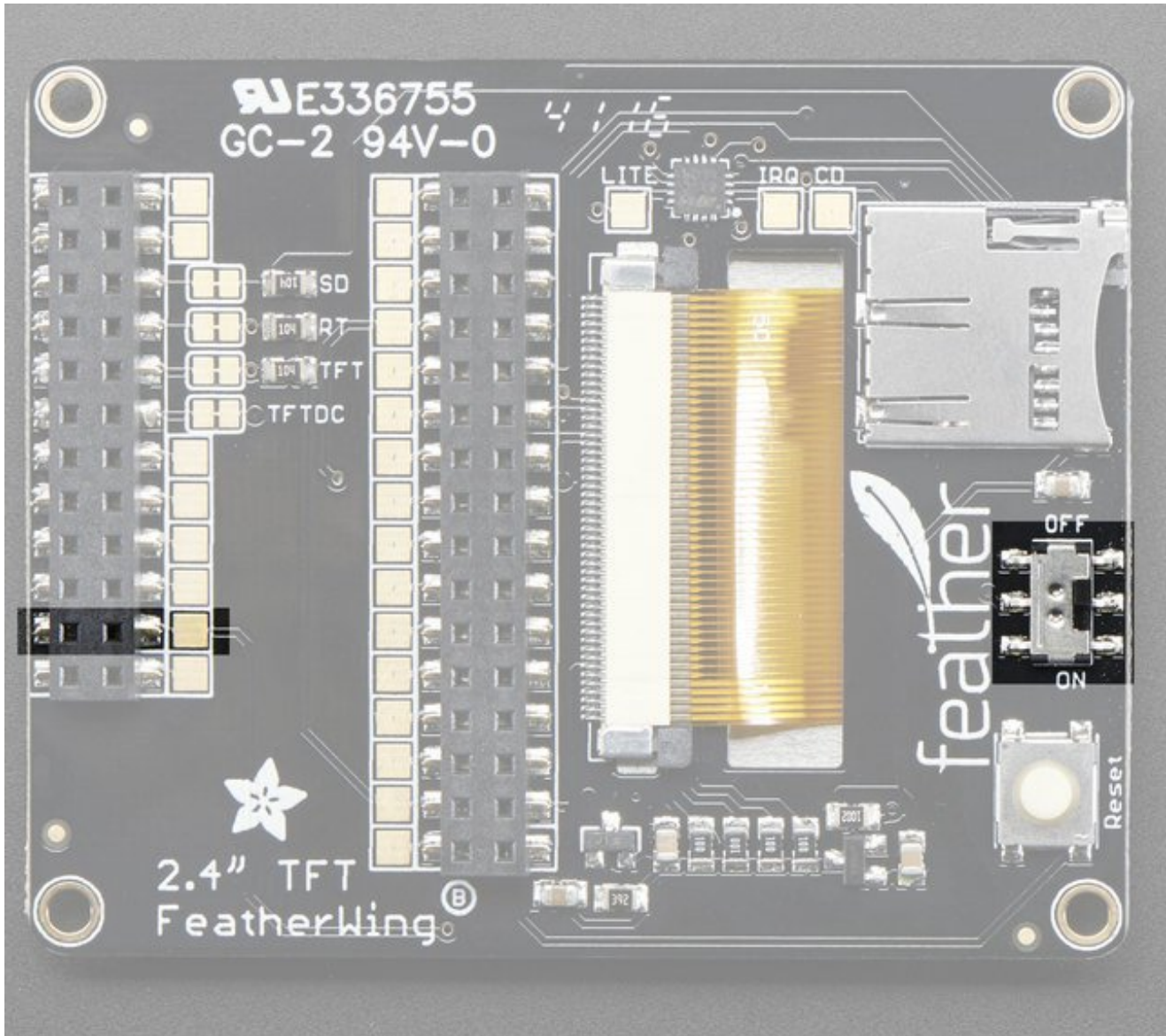
This makes it really easy to use, but a little different to change if you don't want the default setup

Power Pins



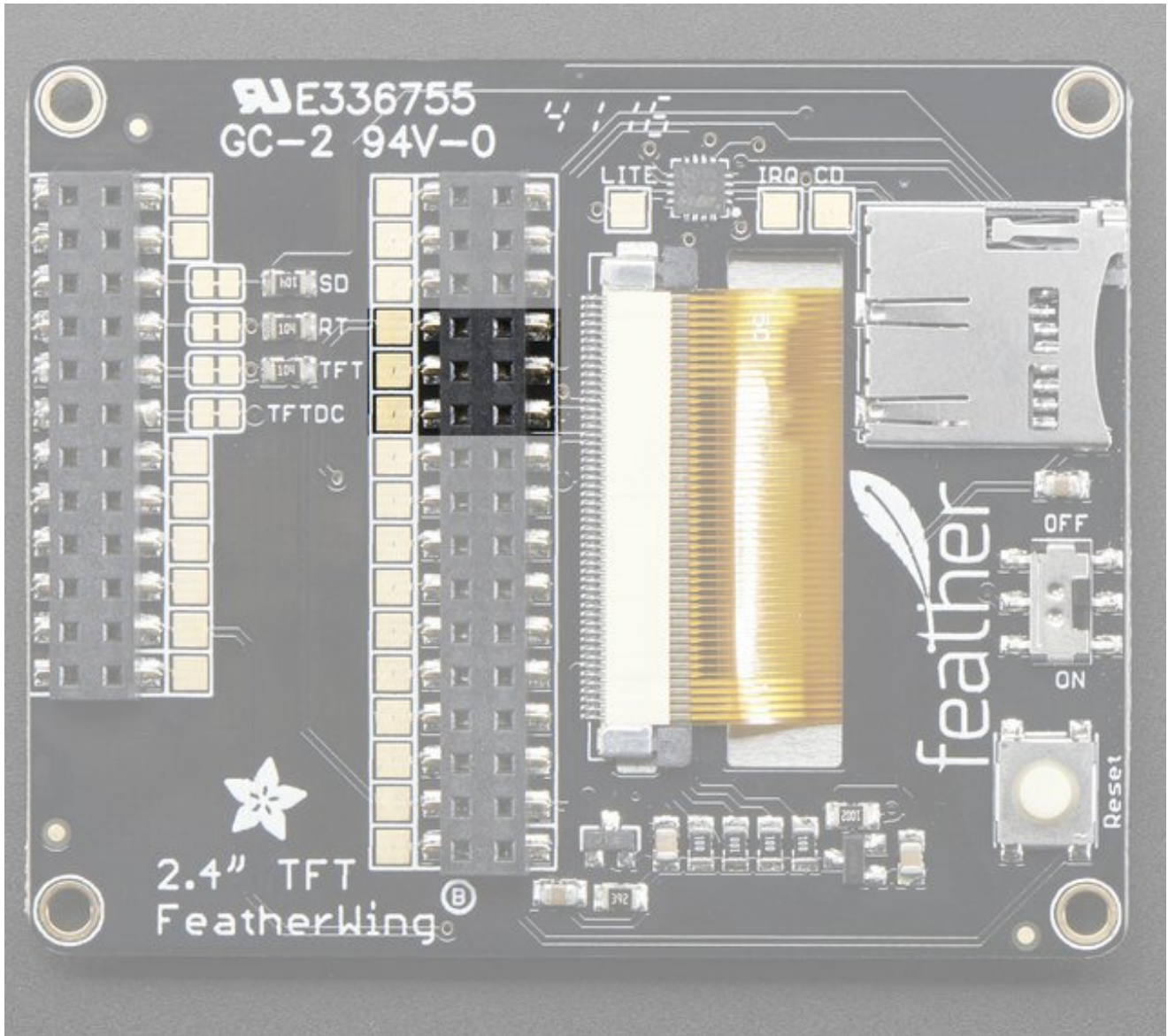
All power for the FeatherWing comes from the 5.0V and GND pins. That includes the

All power for the Featherwing comes from the **3.3V** and **GND** pins. That includes the backlight (which can draw up to 100mA)!



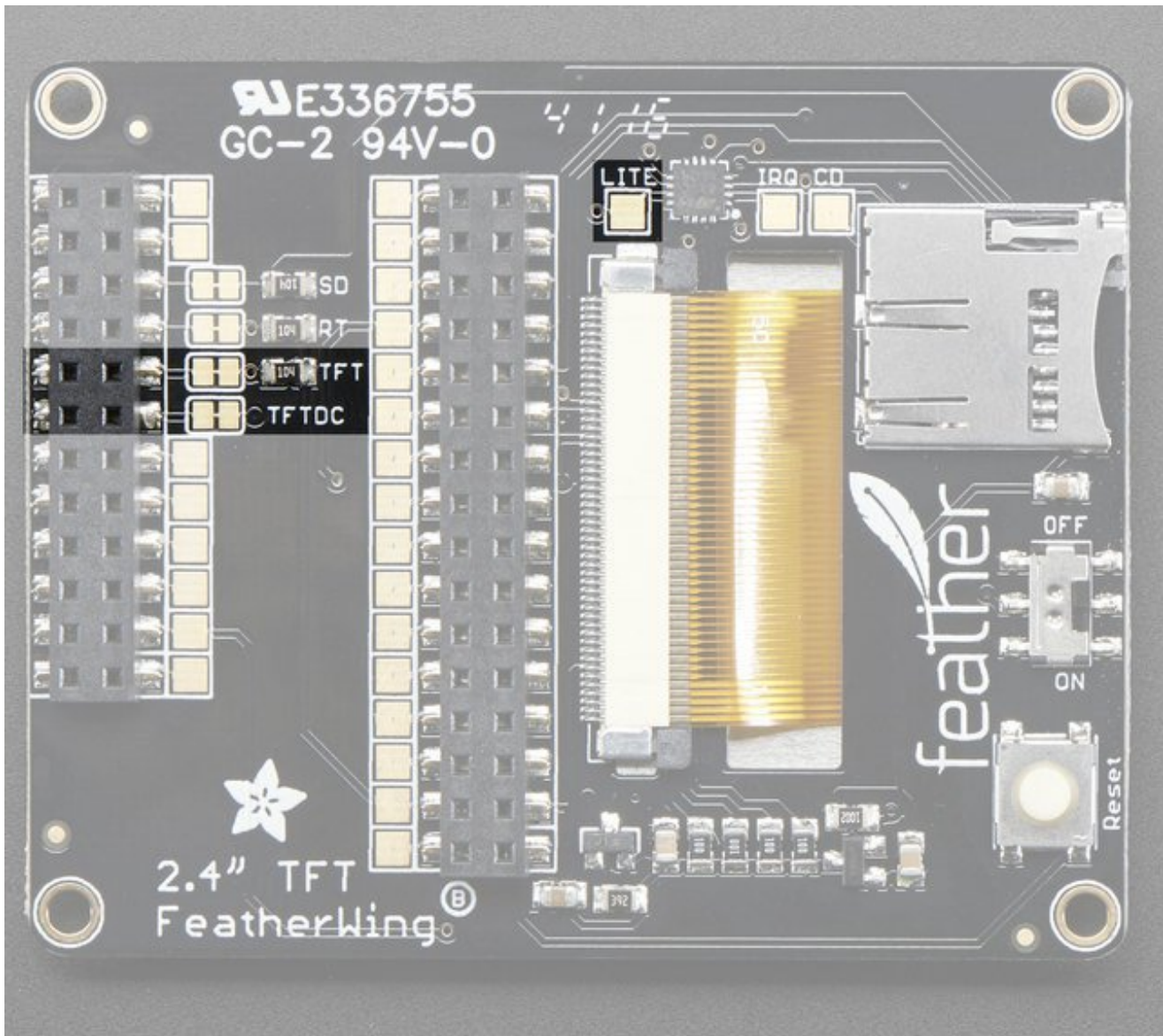
You can turn off the 3.3V power supply with the **EN** pin or the switch attached to that pin. Note that on the Teensy 3x Feather Adapter, this pin doesn't do anything and on the FONA feather, this will not disable the VBAT power supply which is used to power the cellular module

SPI Pins



The TFT display, SD card and touch screen use the SPI interface to communicate. That means **MISO**, **MOSI** and **SCK** are used whenever either are accessed.

TFT Control Pins



In addition, for the TFT display there are **D/C** (Data/Command) and **CS** (Chip Select) pins. These are used to select the display and tell it what kind of data is being sent. These pins can theoretically be changed by cutting the jumper trace and soldering a small wire from the right-hand pad to the pin you'd like to use.

On the **ESP8266**, TFT_CS is pin #0, TFT_DC is pin #15

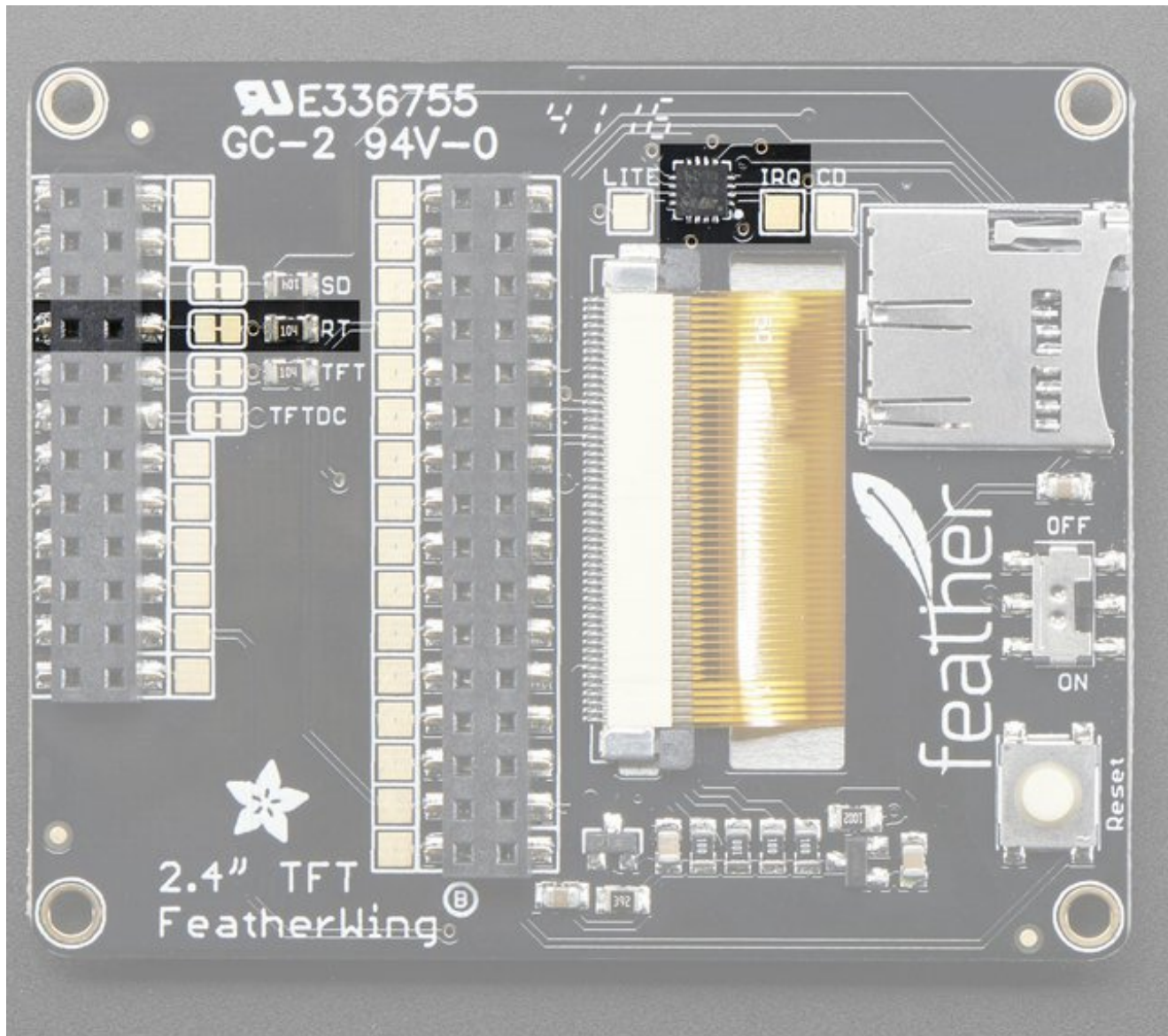
On the **Atmega32u4 or M0** Feather, TFT_CS is pin #9, TFT_DC is pin #10

On the **Teensy Feather**, TFT_CS is pin #4, TFT_DC is pin #10

On the **WICED Feather**, TFT_CS is PA15 and TFT_DC is PB4

There is also **LITE** pin which is not connected to any pads but you can use to control the backlight. Pull low to turn off the backlight. You can connect it to a PWM output pin.

Touch Screen control pins



The touch screen also has a Chip Select line, labeled **RT**. This pin can theoretically be changed by cutting the jumper trace and soldering a small wire from the right-hand pad to the pin you'd like to use.

On the **ESP8266**, RT is pin #16

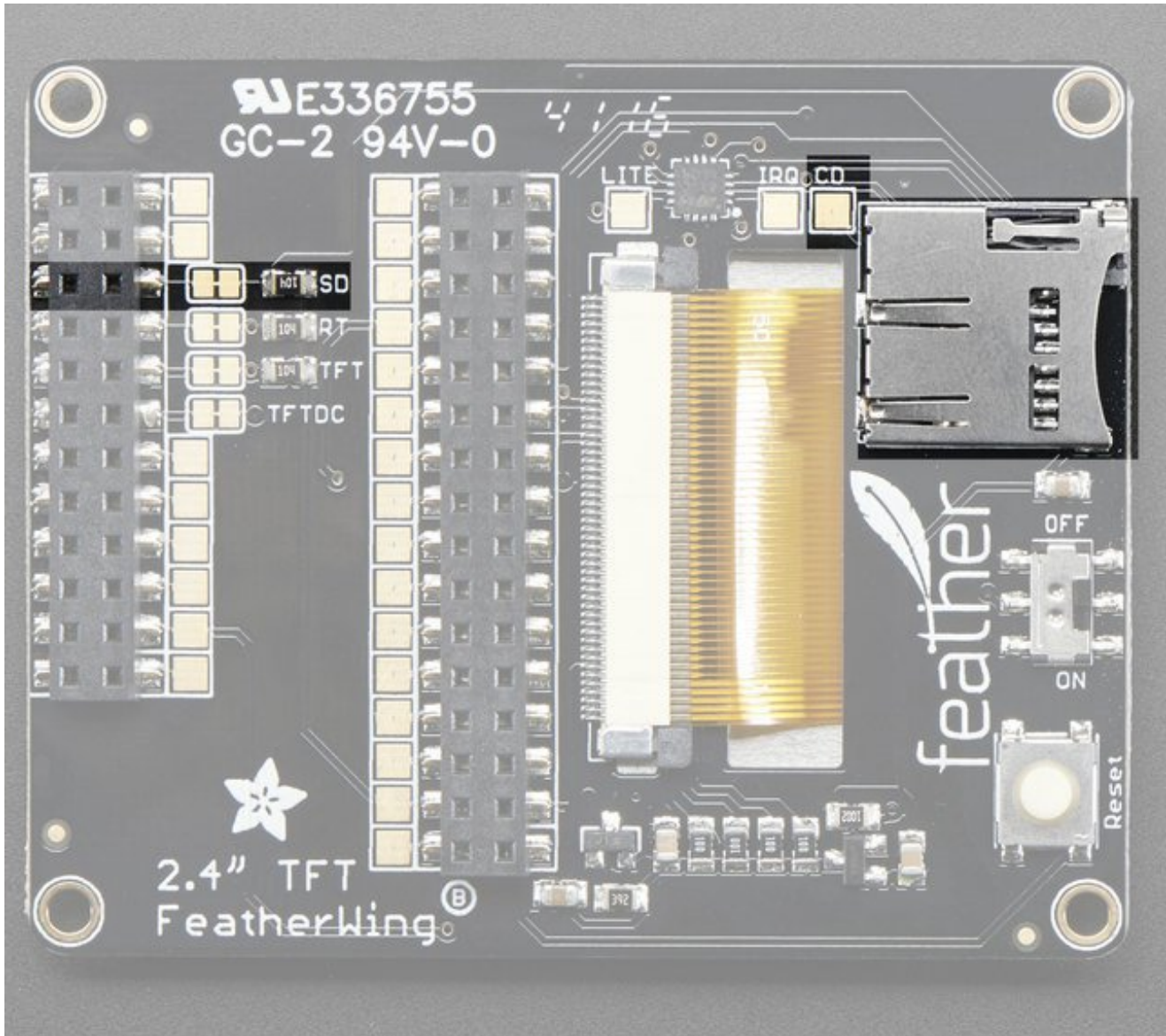
On the **Atmega32u4** or **M0** Feather, RT is pin #6

On the **Teensy Feather**, RT is pin #3

On the **WICED Feather**, RT is PC7

There is also an **IRQ** pin which is not connected to any pads but you can use to detect when touch events have occurred.

SD Card control pins



The SD Card also has a Chip Select line, labeled **SD**. This pin can theoretically be changed by cutting the jumper trace and soldering a small wire from the right-hand pad to the pin you'd like to use.

On the **ESP8266**, SD is pin #2

On the **Atmega32u4** or **M0** Feather, SD is pin #5

On the **Teensy Feather**, SD is pin #8

On the **WICED Feather**, SD is PC5

There is also an **Card Detect (CD)** pin which is not connected to any pads but you can use to detect when a microSD card has been inserted have occurred. It will be shorted to ground when a card is *not* inserted.

TFT Graphics Test

The TFT FeatherWing is basically a combination of our [2.4" TFT Breakout](http://adafru.it/sjD) (<http://adafru.it/sjD>) with the [STMPE610 resistive touch-screen breakout attached](http://adafru.it/1571) (<http://adafru.it/1571>).

Install Libraries

You'll need a few libraries to use this FeatherWing!

Install Adafruit ILI9341 TFT Library

We have example code ready to go for use with these TFTs.

Two libraries need to be downloaded and installed: first is the [Adafruit_ILI9341 library](http://adafru.it/d4d) (<http://adafru.it/d4d>) (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_ILI9341 Library](http://adafru.it/d4e)

<http://adafru.it/d4e>

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

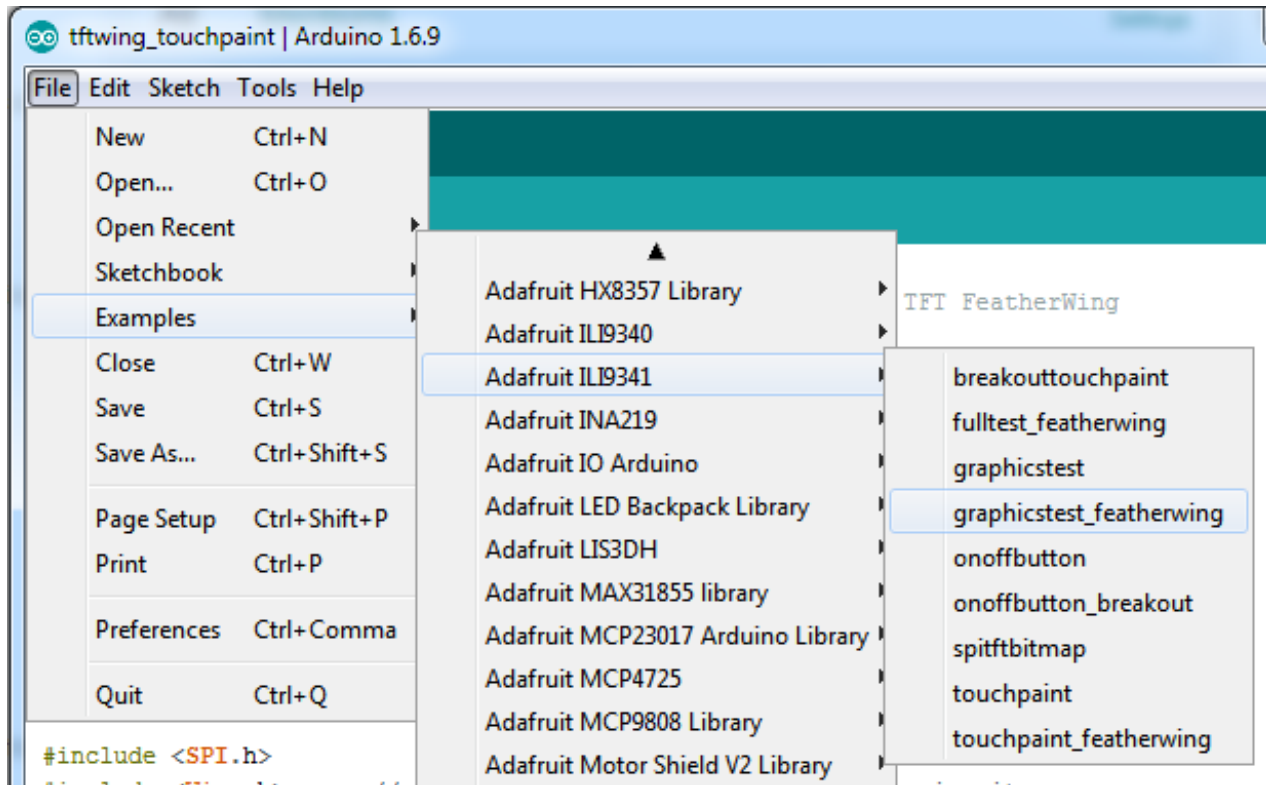
<http://adafru.it/cBB>

Download both ZIP files, uncompress and rename the folders to **Adafruit_ILI9341** (contains **Adafruit_ILI9341.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

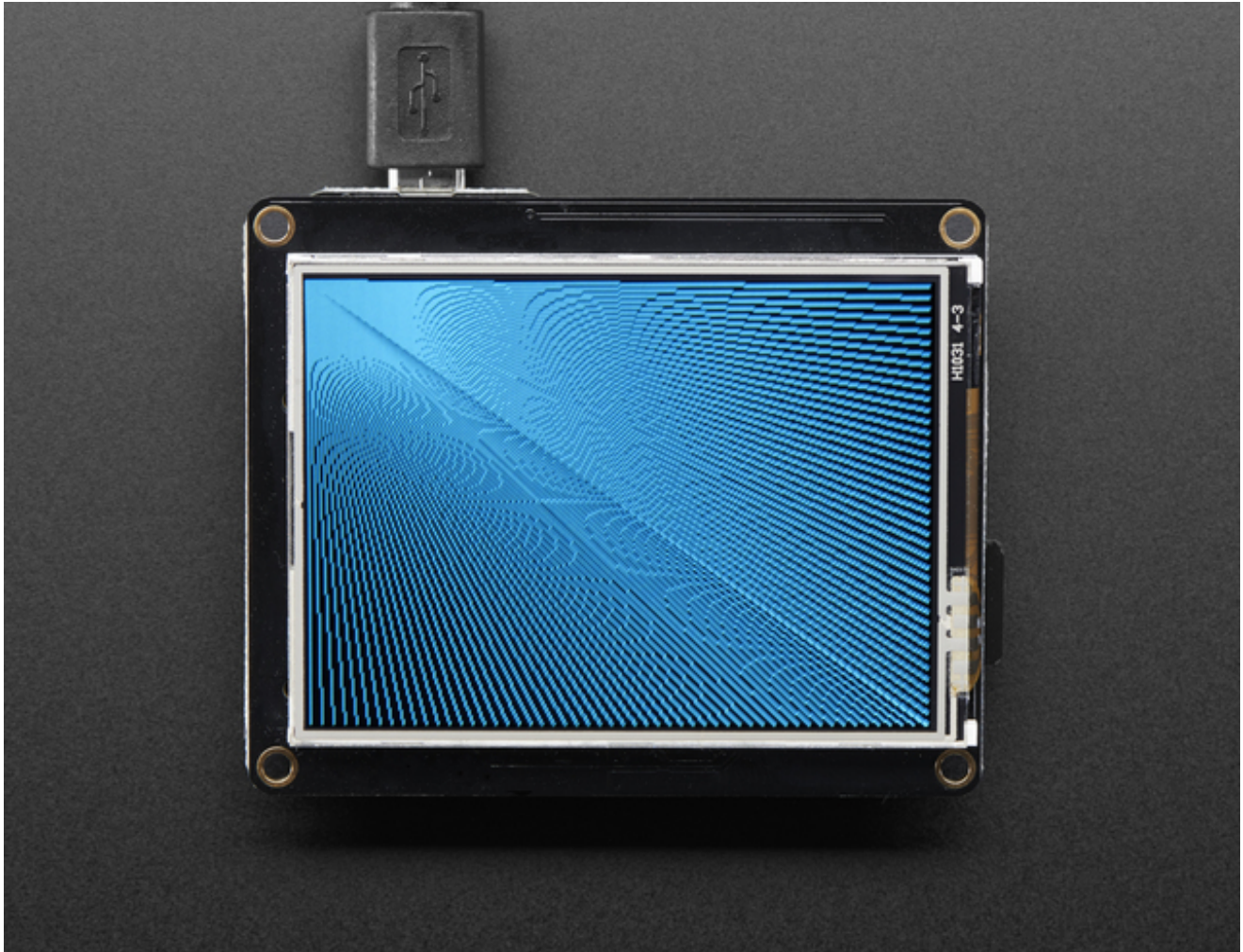
Basic Graphics Test

After restarting the Arduino software, you should see a new **example** folder called **Adafruit_ILI9341** and inside, an example called **graphicstest_featherwing**.

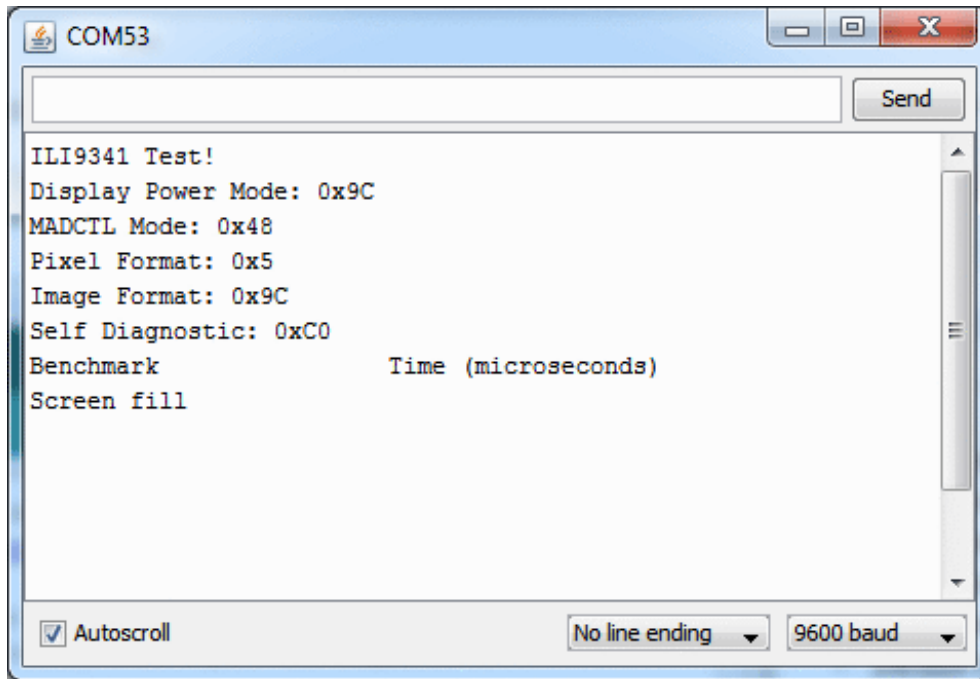


Upload that sketch to your Feather. You should see a collection of graphical tests draw out on the TFT.



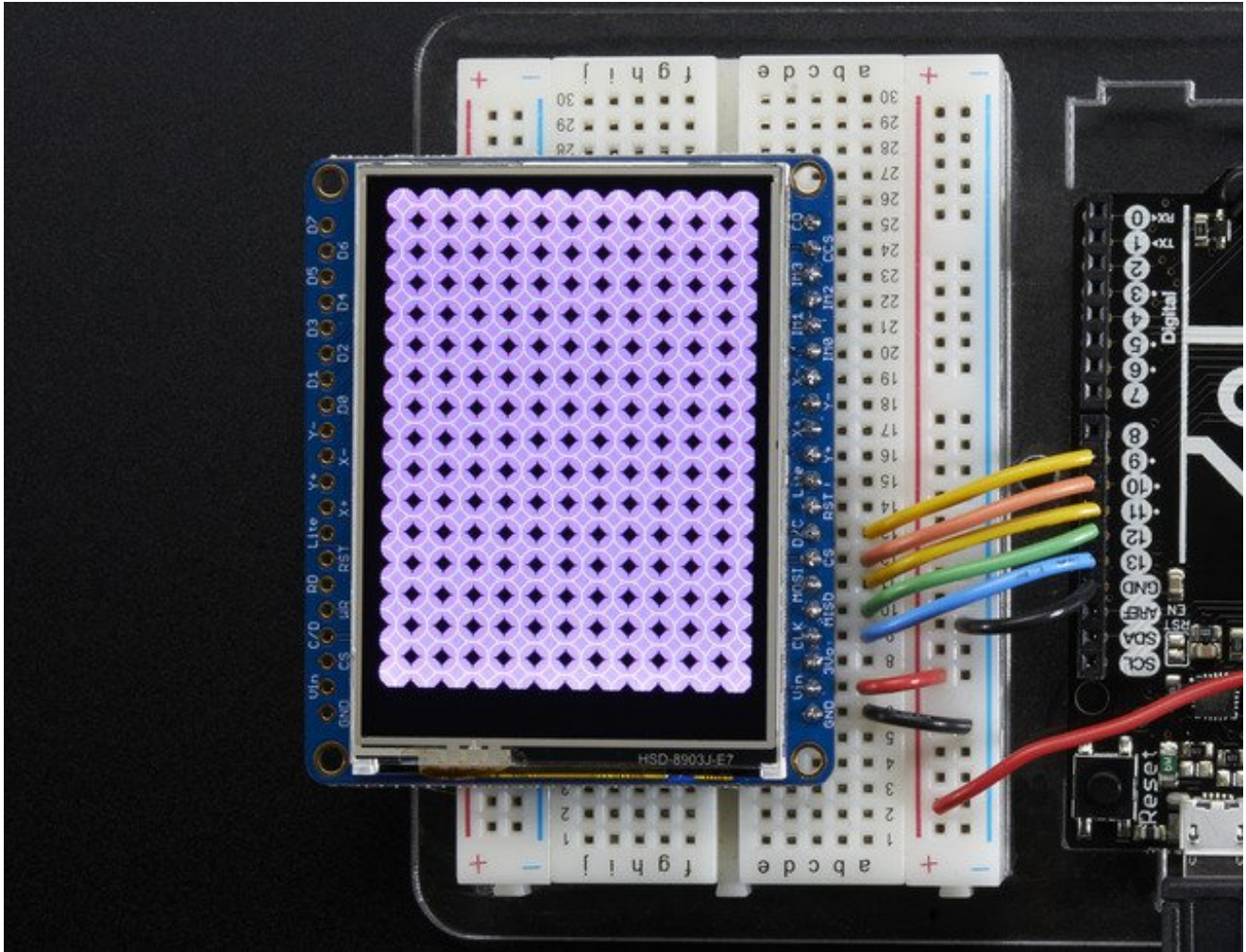


If you're having difficulties, check the serial console. The first thing the sketch does is read the driver configuration from the TFT, you should see the same numbers as below. That will help you determine if the TFT is found, if not, check your Feather soldering!



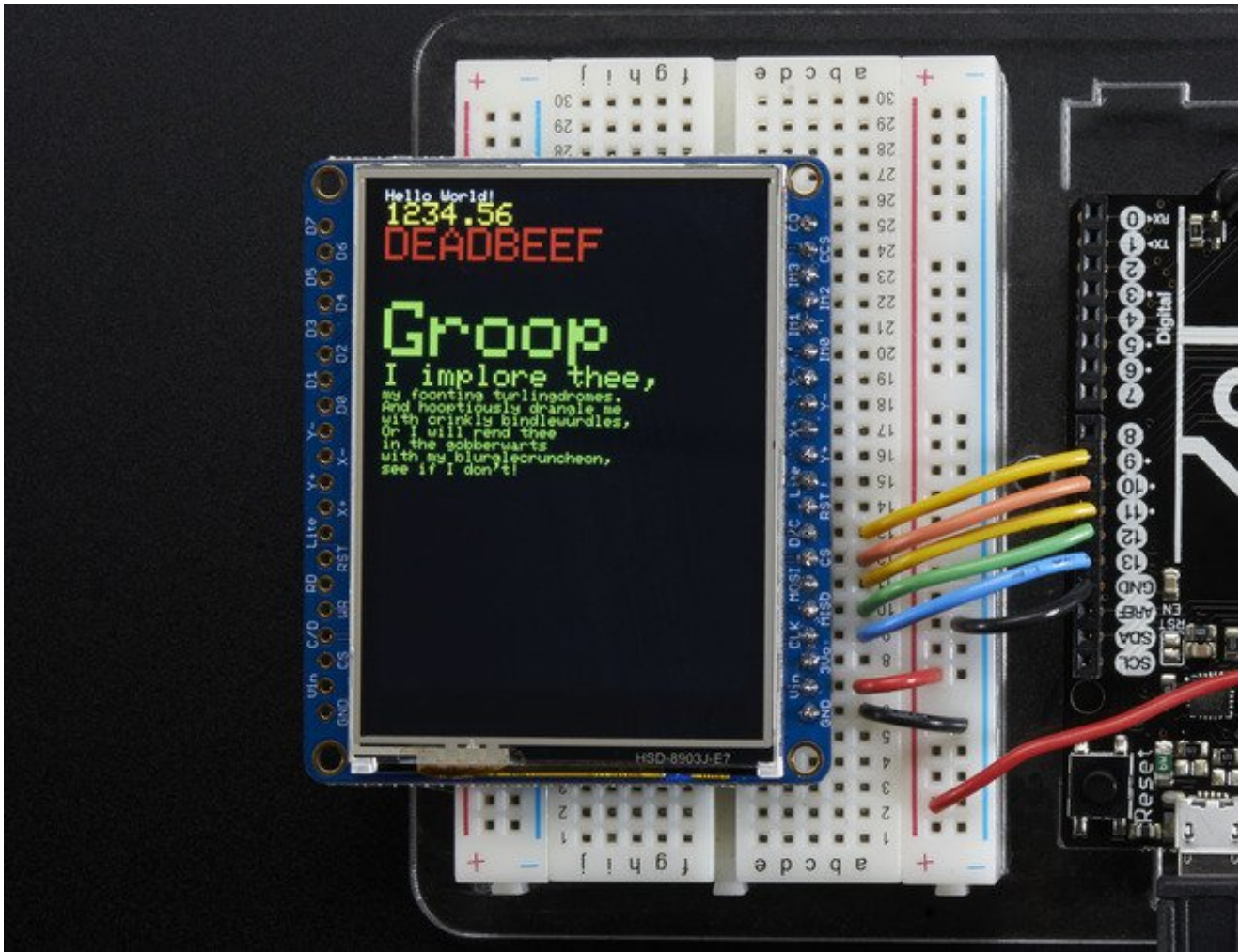
Once you've got the demo working, check out the detailed documentation over at <http://learn.adafruit.com/adafruit-gfx-graphics-library> (<http://adafru.it/aPx>) for more information on how to use the GFX library to draw whatever you like!

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 here <http://learn.adafruit.com/adafruit-gfx-graphics-library> (<http://adafru.it/aPx>)

It covers the latest and greatest of the GFX library. The GFX library is used in both 8-bit and SPI modes so the underlying commands (drawLine()) for example) are identical!



Resistive Touch Screen

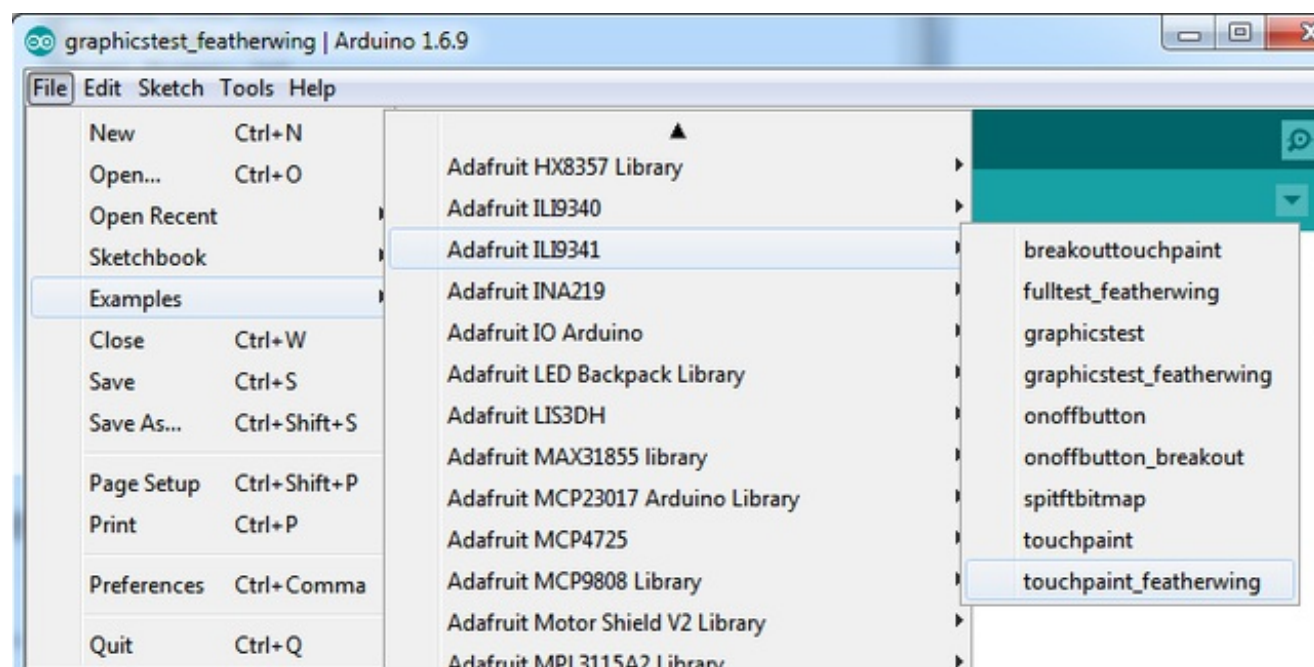
The LCD has a 2.8" 4-wire resistive touch screen glued onto it. You can use this for detecting finger-presses, stylus', etc. Normally, you'll need 4 pins to talk to the touch panel **but** we decided to go all snazzy and put a dedicated touch screen driver onto the shield. The driver shares the SPI pins with the TFT and SD card, so only one extra pin is needed. 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 \(http://adafru.it/d4f\)](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.

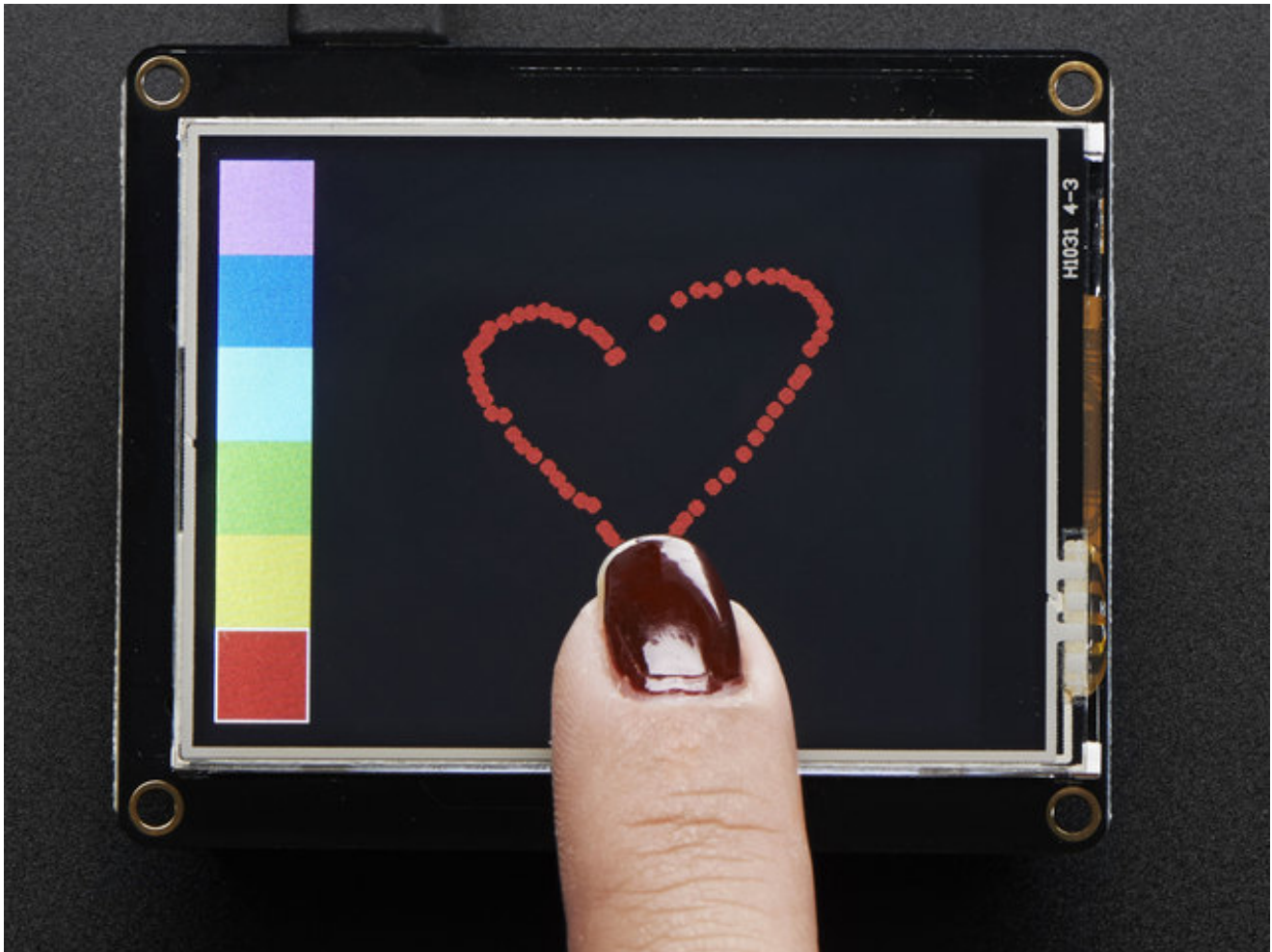
[Download the Adafruit STMPE610 Library](http://adafru.it/d4g)
<http://adafru.it/d4g>

Touchscreen Paint Demo

Now that you've got the basic TFT graphics demo working, let's add in the touchscreen. Run and upload the **touchpaint_featherwing** demo



Upload to your Feather and have fun!



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. 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 Feather soldered right and the correct CS pin!

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)



Save and rename it to **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->fulltest_featherwing** 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(bitmapfilename, x, y);
```

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

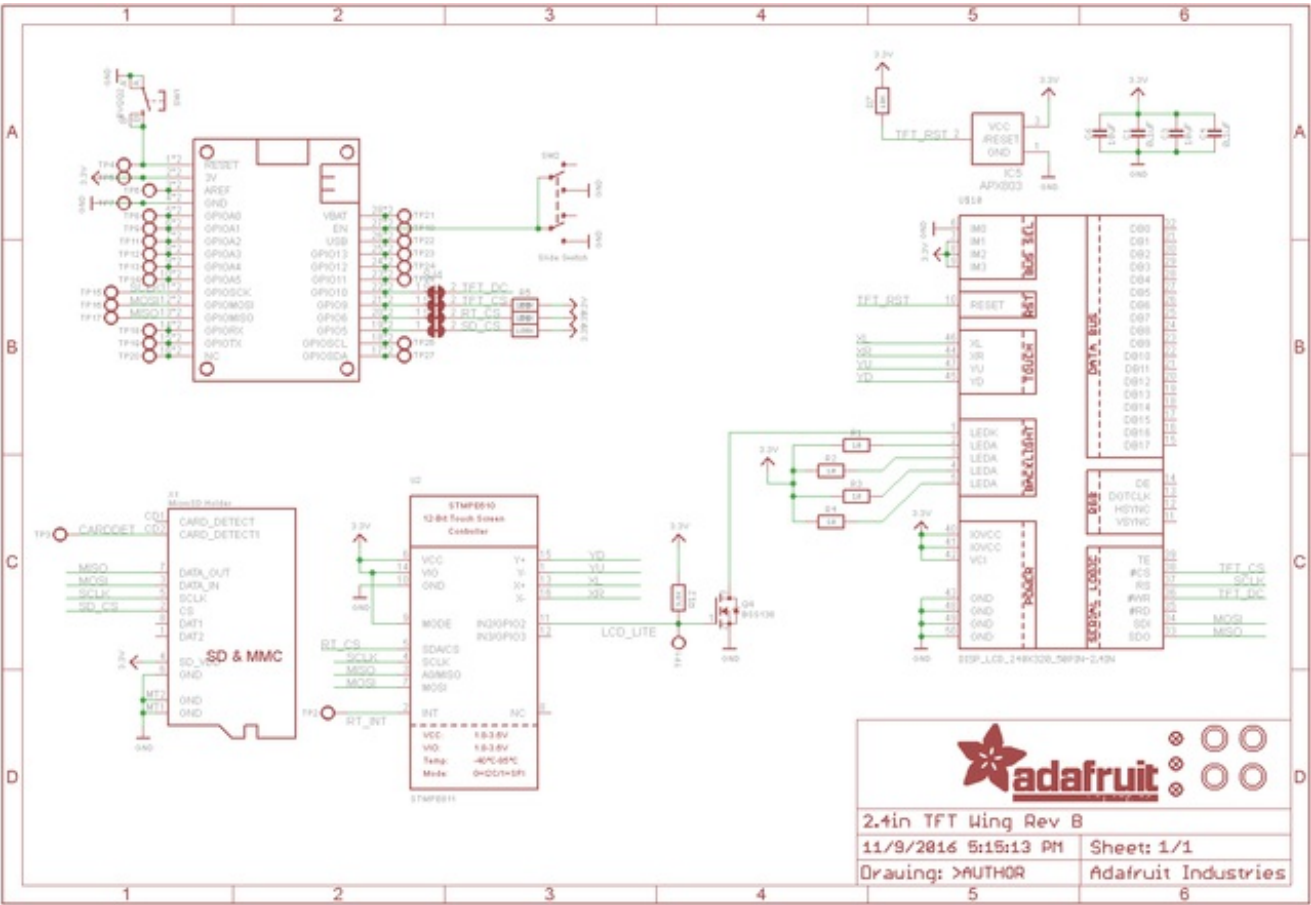


Downloads

Datasheets & More

- [Fritzing object in Adafruit Fritzing Library \(http://adafru.it/c7M\)](http://adafru.it/c7M)
- [STMPE610 \(http://adafru.it/d4k\)](http://adafru.it/d4k)
- [ILI9341 \(TFT controller\) \(http://adafru.it/d4l\)](http://adafru.it/d4l)
- [EagleCAD PCB files on GitHub \(http://adafru.it/ska\)](http://adafru.it/ska)

Schematic



Fabrication Print

