

## Adafruit 3.5" 480x320 TFT FeatherWing

Created by lady ada



Last updated on 2018-01-14 02:13:04 AM UTC

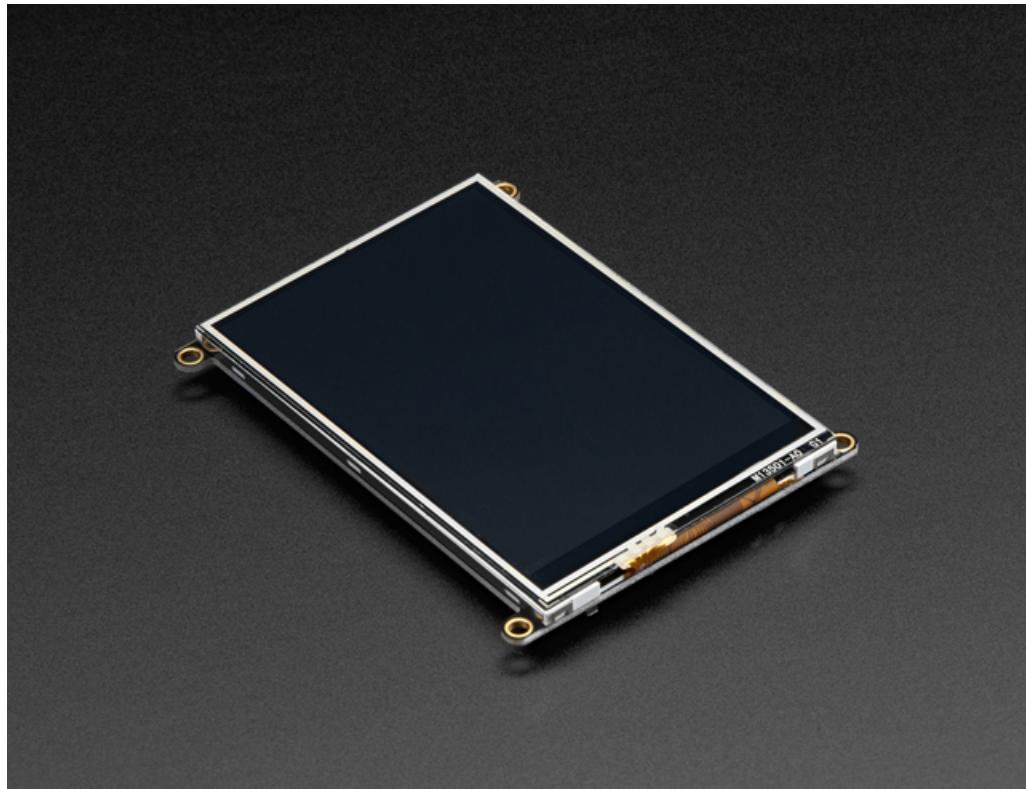
## Guide Contents

Guide Contents	2
Overview	3
Pinouts	6
Power Pins	6
SPI Pins	6
Touch Screen control pins	7
SD Card control pins	8
TFT Graphics Test	9
Install Libraries	9
Install Adafruit HX8537D TFT Library	9
Basic Graphics Test	10
Adafruit GFX Library	13
Resistive Touch Screen	15
Touchscreen Paint Demo	15
Drawing Bitmaps	18
Downloads	20
Datasheets & More	20

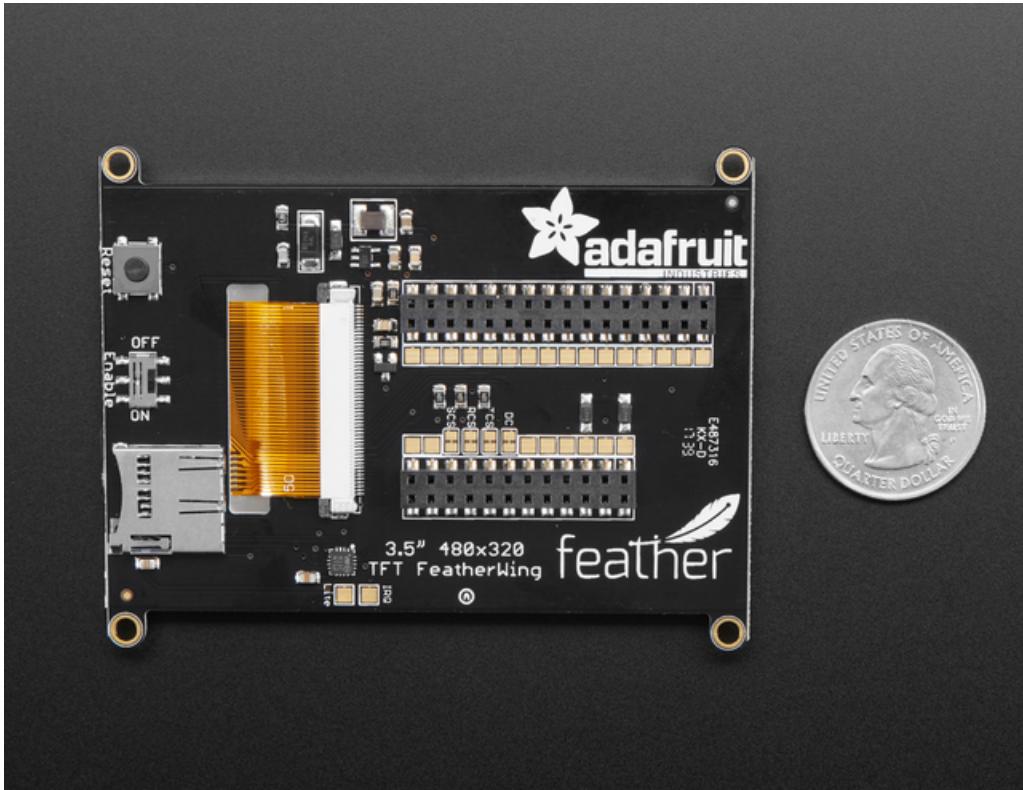
## Overview

---

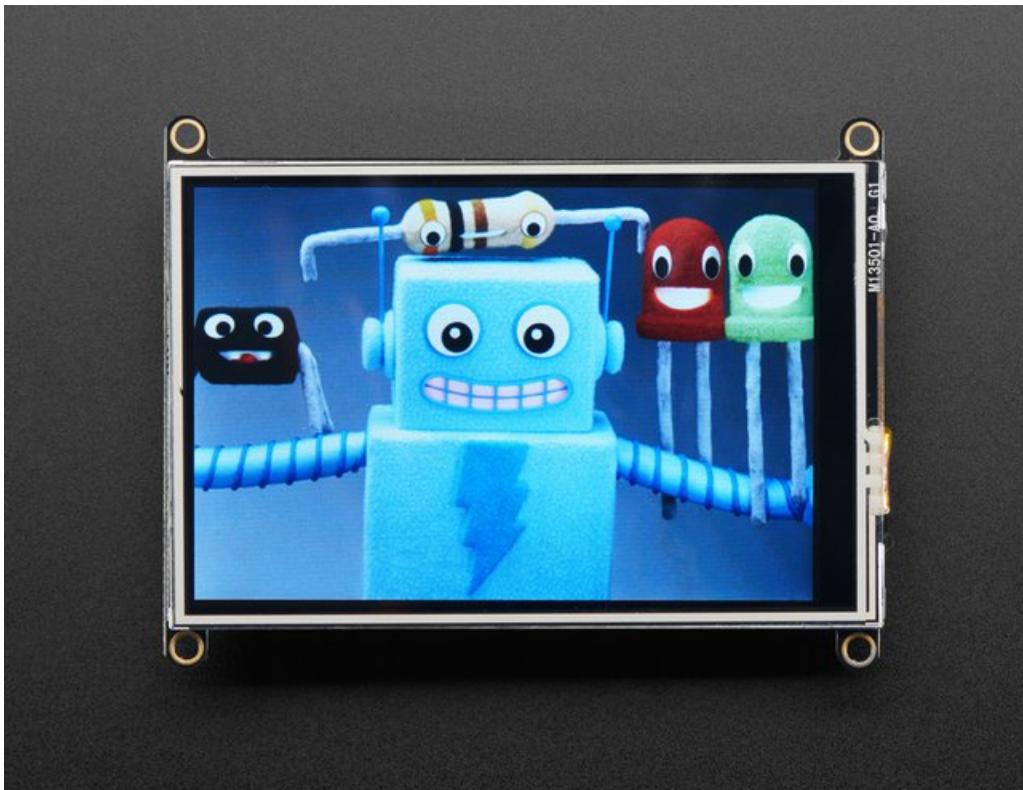
A Feather board without ambition is a Feather board without FeatherWings! Spice up your Feather project with a beautiful 3.5" touchscreen display shield with built in microSD card socket. This TFT display is 3.5" diagonal with a bright 6 white-LED backlight. You get a massive 480x320 pixels with individual 16-bit color pixel control. It has way more resolution than a black and white 128x64 display, and twice as much as our 2.4" TFT FeatherWing. 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. **It works with any and all Feathers** but given the large display it works *best* with our faster boards like the ESP8266, ESP32, M0, WICED, and Teensy. 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.





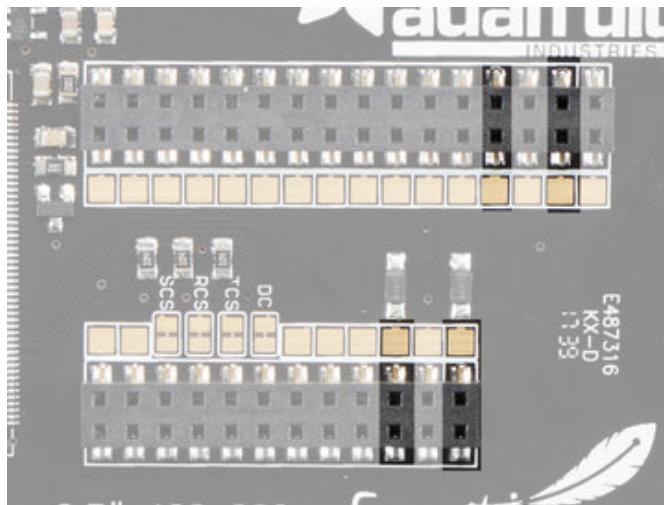
## Pinouts

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

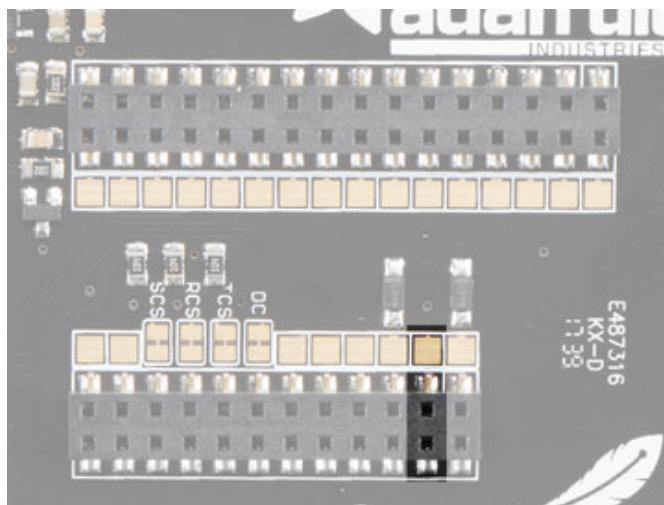
The FeatherWing has an **ENABLE** switch which will disable both Feather and Wing - so make sure it is turned ON while you're using it

## Power Pins



The 3.5" TFT FeatherWing has a really beefy backlight, and we don't want to overwhelm the 3.3V regulator on the Feather so we use two Schottkey diodes to take the higher of **VBAT** and **VUSB** to feed into the backlight boost converter.

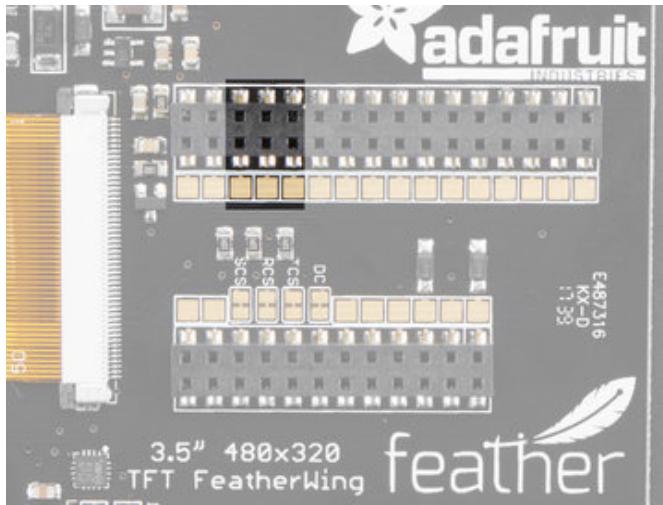
The **GND** and **3.3V** pins are used for powering the rest of the device such as the TFT and touchscreen controller



You can turn off the 3.3V power supply with the **EN** pin or the switch attached to that pin. The enable/disable switch on the bottom of the Wing will also deactivate the backlight

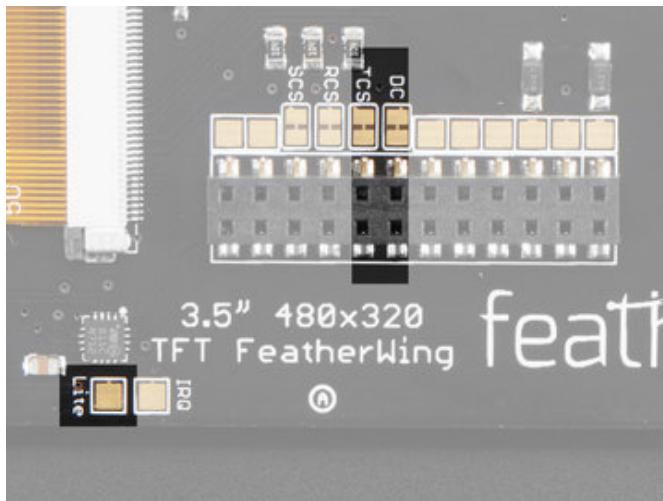
Note that on the Teensy 3x Feather Adapter, this pin doesn't do anything and on the FONA feather, this will not disable the VBAT power supply which is used to power the cellular module.

## 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.

*(Ooops the highlighted pins are off by one, they should be shifted to the right one pin!)*



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 farther-from-the-socket-header pad to the pin you'd like to use.

On the **ESP8266**, **TFT\_CS** is pin #0, **TFT\_DC** is pin #15

On the **ESP32**, **TFT\_CS** is pin #15, **TFT\_DC** is pin #33

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

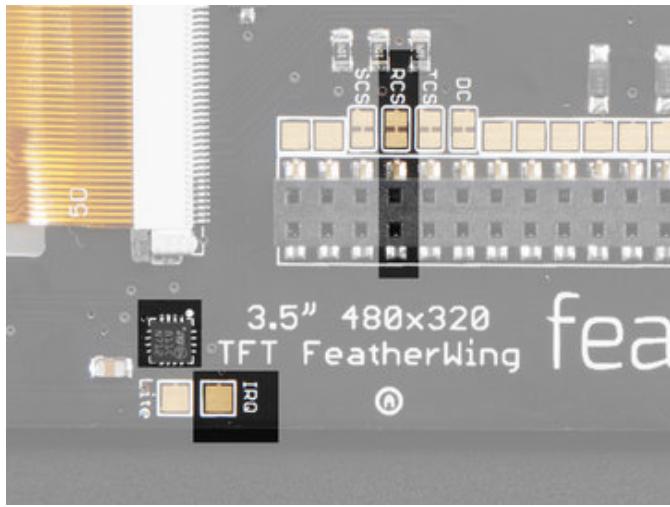
On the **WICED Feather**, **TFT\_CS** is PA15 and **TFT\_DC** is PB4

On the **nRF52 Feather**, **TFT\_CS** is #31 and **TFT\_DC** is #11

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.

Note: Pin 9 is used for communication with the SIM800 chip on the Feather Fona. You will have to remap pin 9 to an unused pin when using with a Feather Fona.

## 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 **ESP32**, RT is pin #32

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

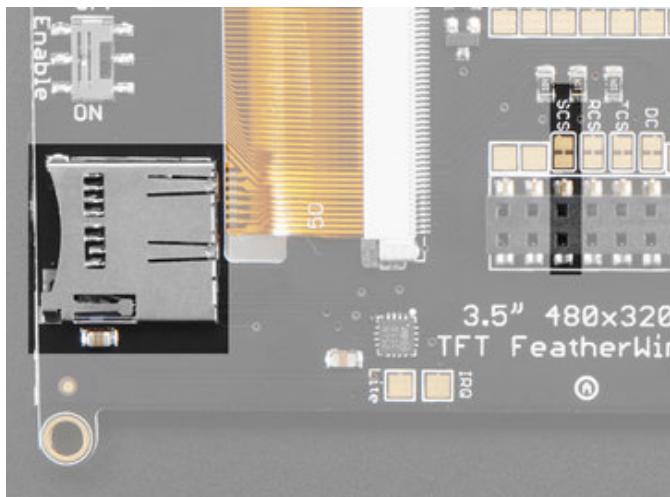
On the **Teensy Feather**, RT is pin #3

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

On the **nRF52 Feather**, RT is #30

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 **ESP32** SD is pin #14

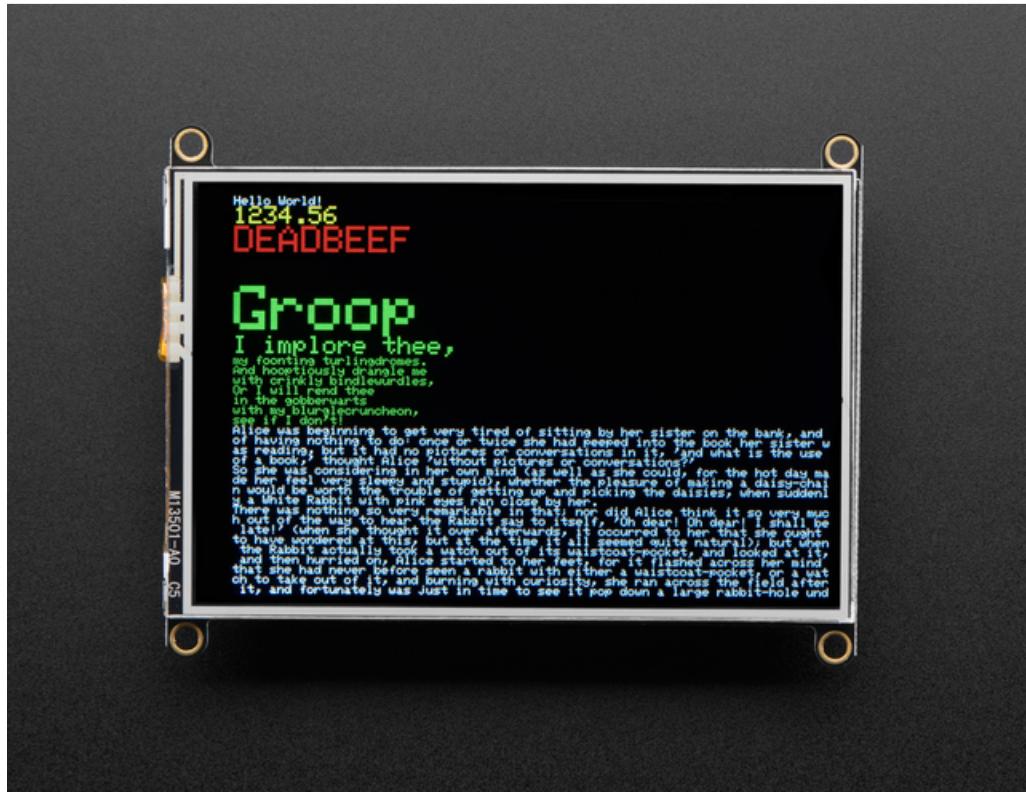
On the **Atmega32u4 or M0 Feather**, SD is pin #5

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

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

On the **nRF52 Feather**, SD is pin #27

## TFT Graphics Test



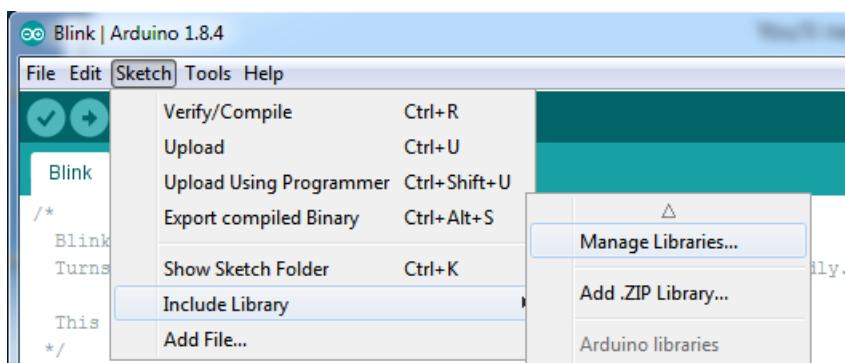
The TFT FeatherWing is basically a combination of our [3.5" TFT Breakout](#) with the [STMP610](#) resistive touch-screen [breakout attached](#).

Before you continue, make sure the FeatherWing's ENABLE switch is turned ON! If it's OFF, the Feather won't work and you will be very confused :)

## Install Libraries

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

From within the Arduino IDE, open up the **Library Manager...**

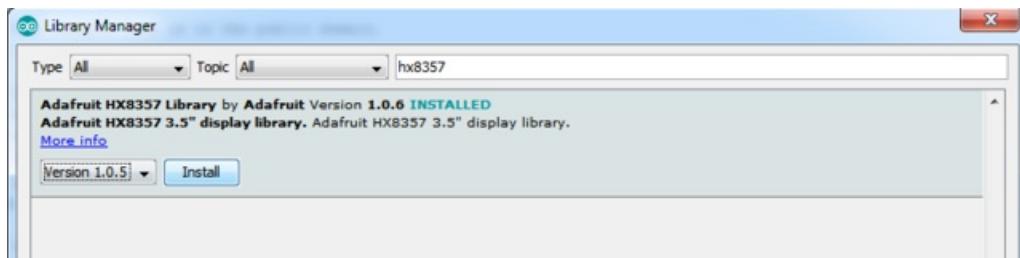


## Install Adafruit HX8537D TFT Library

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

Two libraries need to be downloaded and installed: first is the [Adafruit HX8357 library](#) (this contains the low-level code specific to this device), and second is the [Adafruit GFX Library](#) (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

Search for **HX8357** and install the [Adafruit HX8357 library](#) that pops up!



Next up, search for **Adafruit GFX** and locate the core library. A lot of libraries may pop up because we reference it in the description so just make sure you see **Adafruit GFX Library** in bold at the top.

Install it!

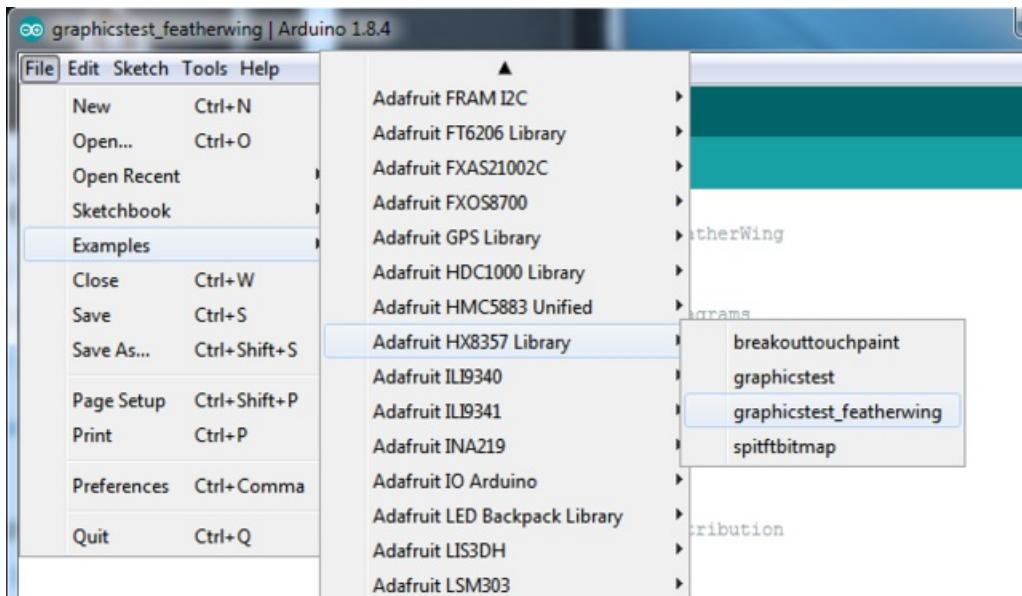


[For more details about this process, we have a tutorial introducing Arduino library concepts and installation](#)

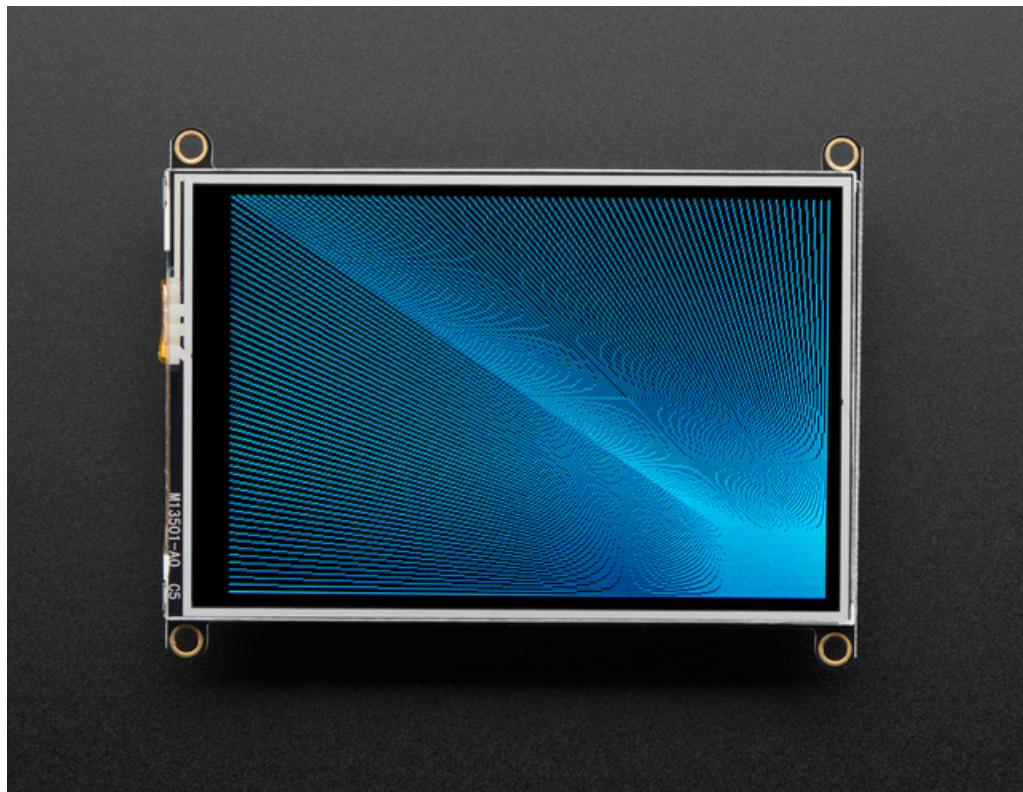
We recommend you restart the IDE

## Basic Graphics Test

After restarting the Arduino software, you should see a new `example` folder called `Adafruit_HX8357` and inside, an example called `graphicstest_featherwing`.



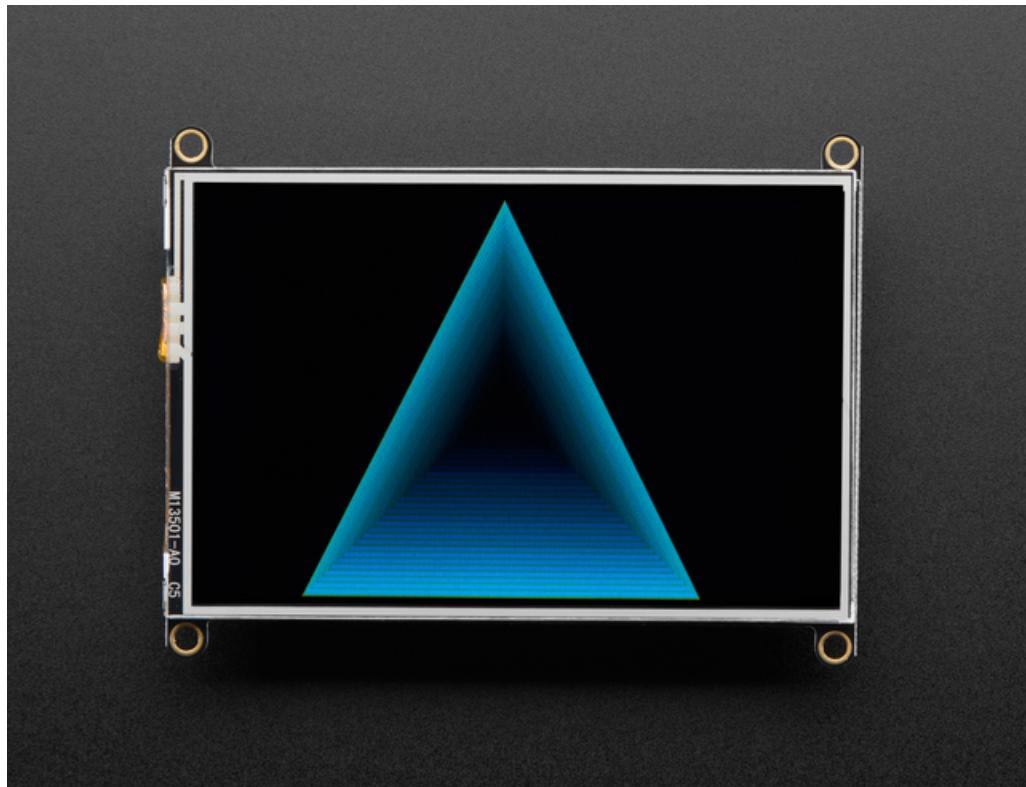
Upload that sketch to your Feather. You should see a collection of graphical tests draw out on the TFT.



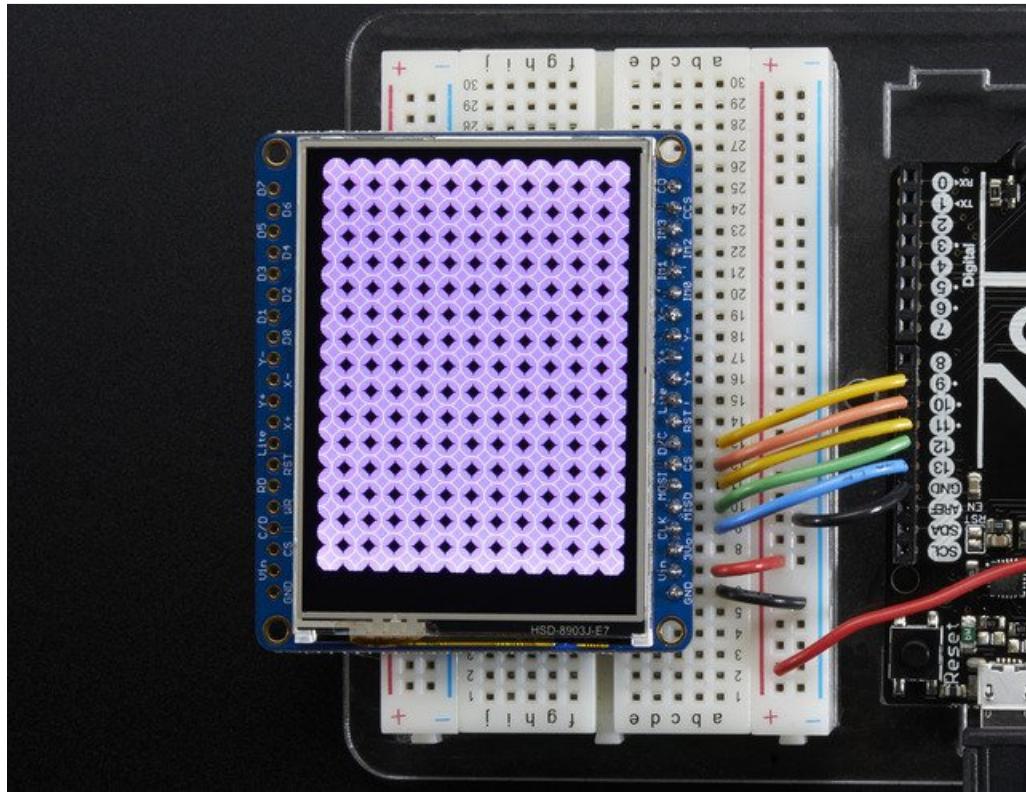
If you're having difficulties, check the serial console. The first thing the sketch does is read the driver configuration from the TFT, you should see the same numbers as below. That will help you determine if the TFT is found, if not, check your Feather soldering!

```
TeensyMonitor: COM129 Online
HX8357D Test!
Display Power Mode: 0x9C
MADCTL Mode: 0xC0
Pixel Format: 0x5
Image Format: 0x0
Self Diagnostic: 0xC0
Benchmark           Time (microseconds)
Text                353731
Lines               1114843
Rectangles (outline) 90918
```

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

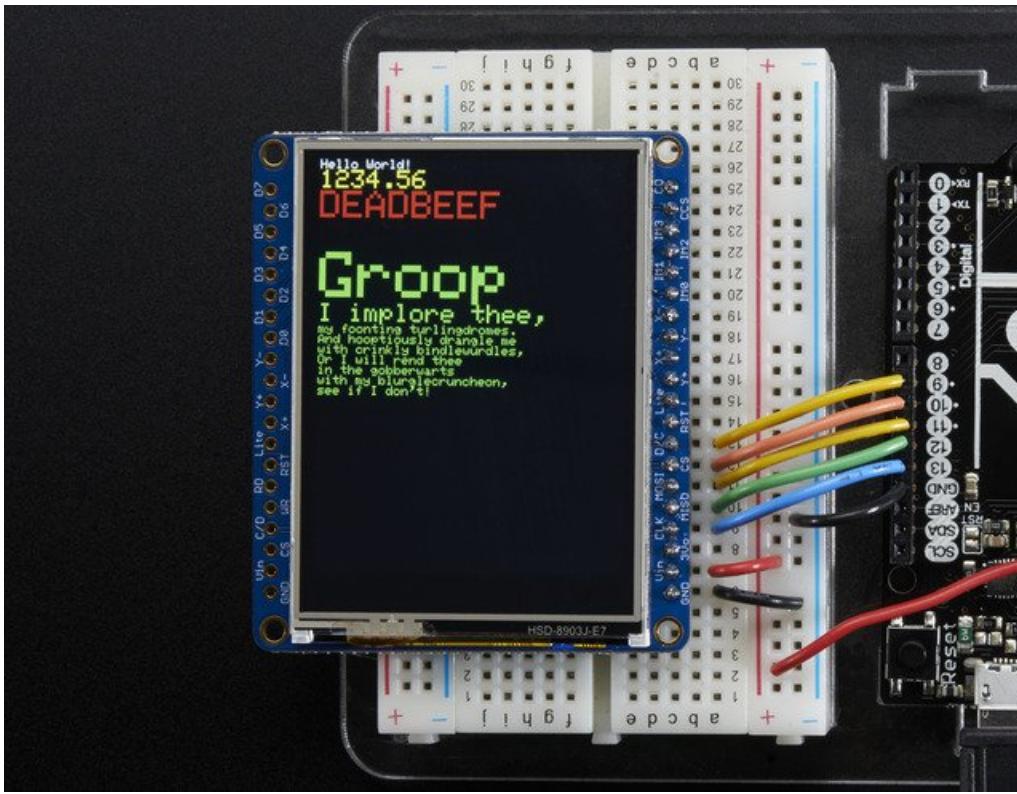


## Adafruit GFX Library



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>

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 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.

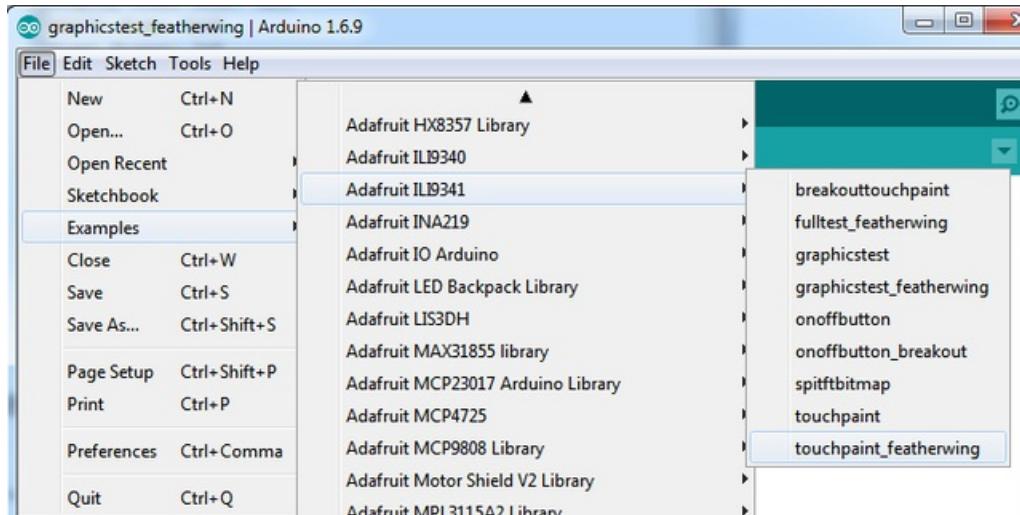
To control the touchscreen you'll need one more library - the STMPE610 controller library which does all the low level chatting with the STMPE610 driver chip. Use the library manager to install the **Adafruit STMPE610** library



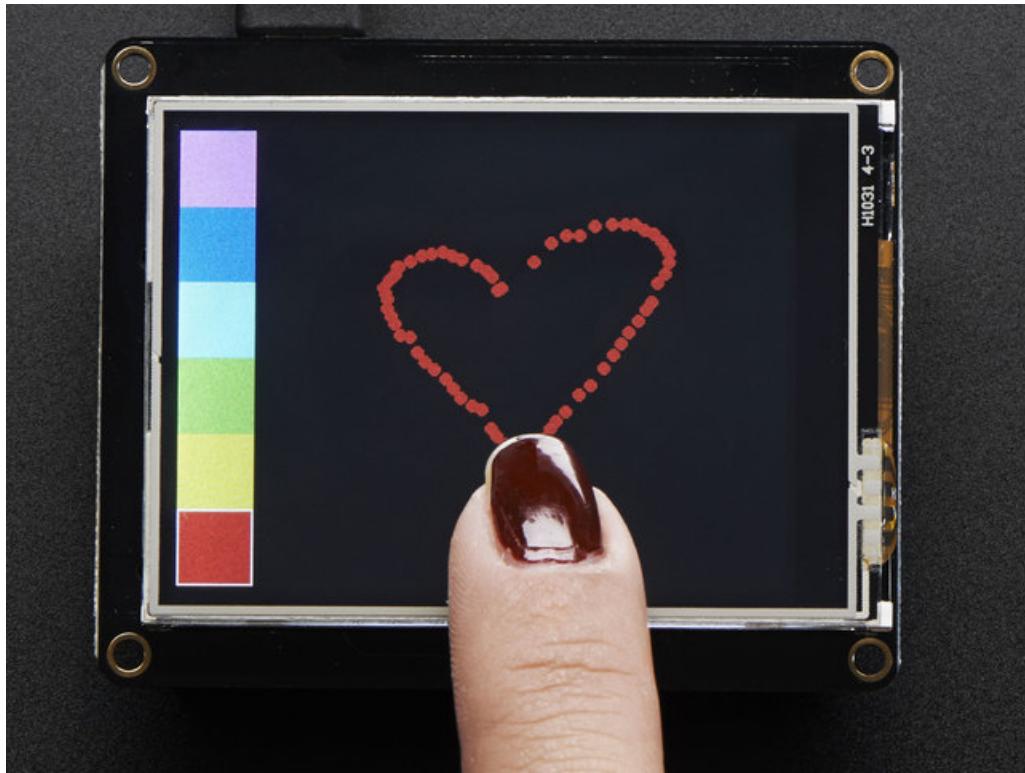
## Touchscreen Paint Demo

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

- If you have the **2.4" TFT Featherwing**, run the **Adafruit ILI9341->touchpaint\_featherwing** demo
- If you have the **3.5" TFT Featherwing**, run the **Adafruit HX8357->touchpaint\_featherwing** demo



Upload to your Feather and have fun!



The touch screen is made of a thin glass sheet, and it's 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

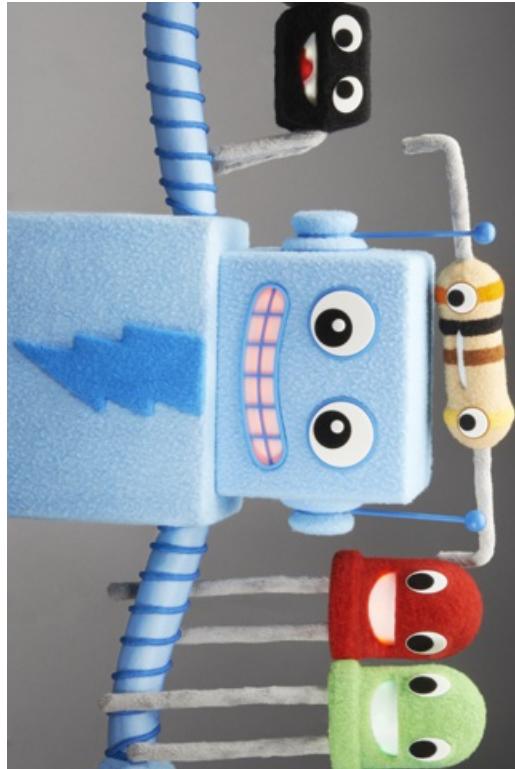
```
if (ts.touched())
```

## Drawing Bitmaps

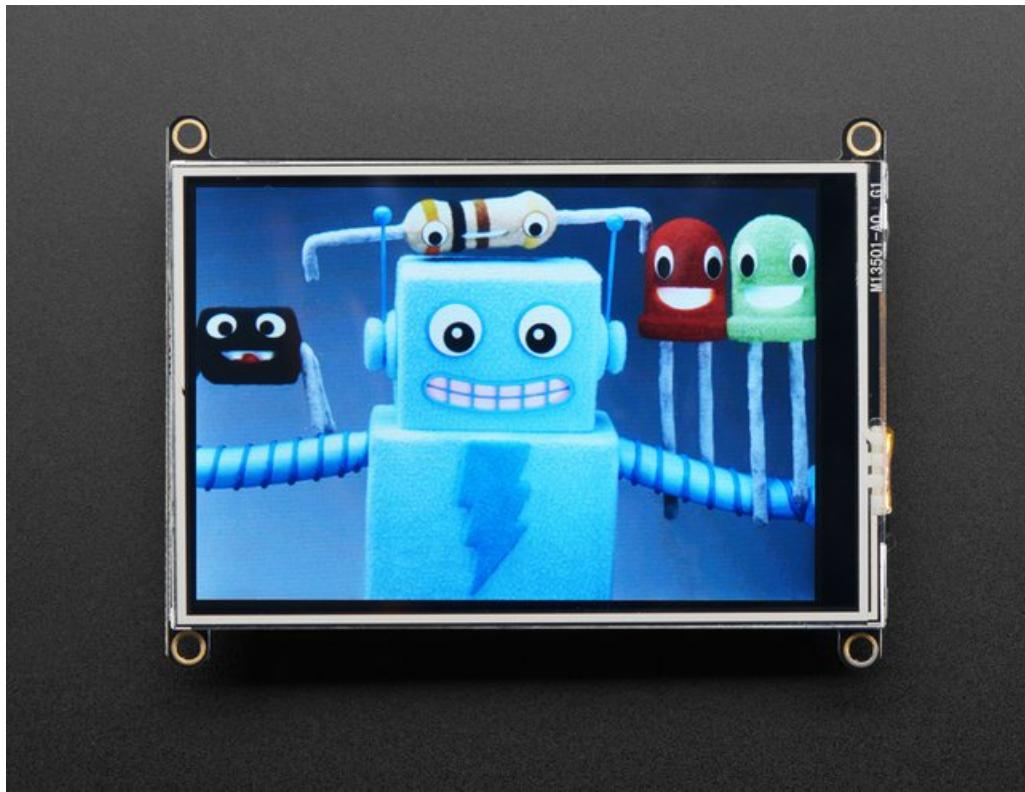
---

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 adabot and friends



Save and rename it to **adabot.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\_HX8357>fulltest\_featherwing** example to your Feather + Wing. You will see your electronic friends appear!



To make new bitmaps, make sure they are less than 480 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 480x240 and placed in any location on the screen.

## Downloads

### Datasheets & More

- [STMPE610 Touch Controller Datasheet](#)
- [Datasheet for the HX8357D chipset controller](#)
- [Datasheet for the 3.5" TFT display \(raw\)](#)
- [EagleCAD PCB files on GitHub](#)
- [Fritzing object in Adafruit Fritzing library](#)

