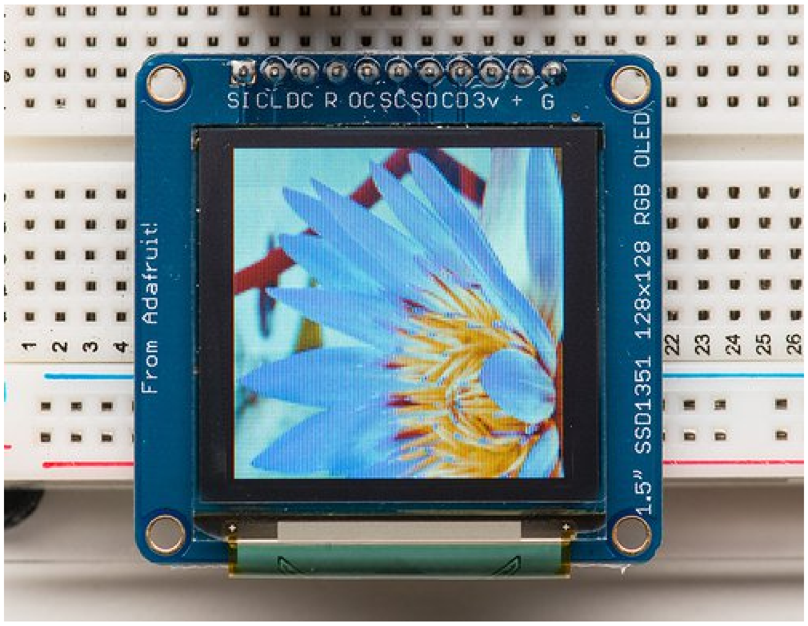
adafruit learning system

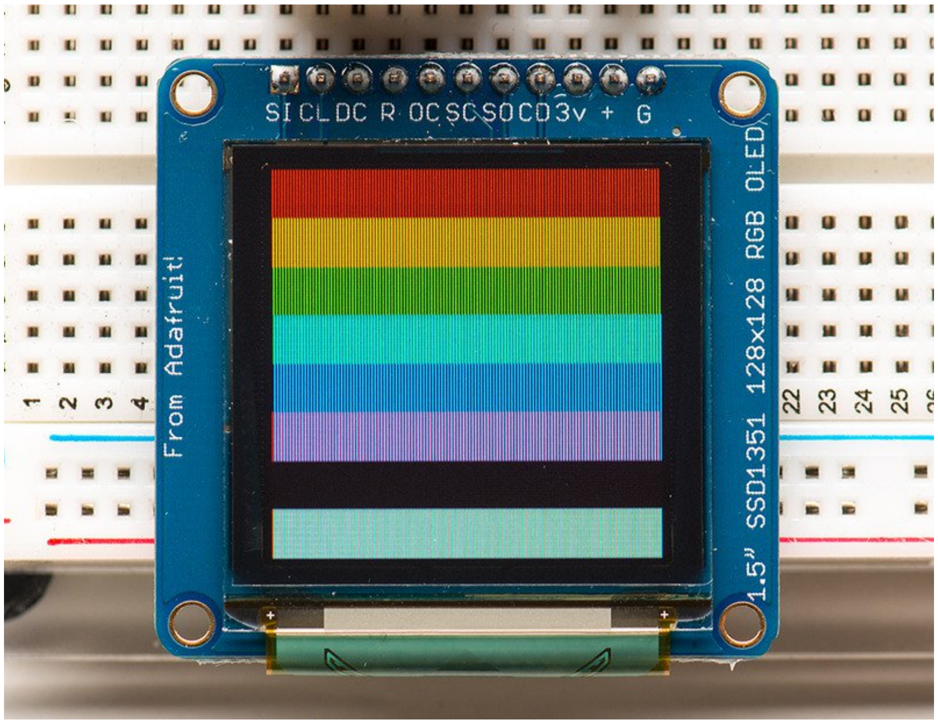
Adafruit 1.27" and 1.5" Color OLED Breakout Board

Created by Bill Earl



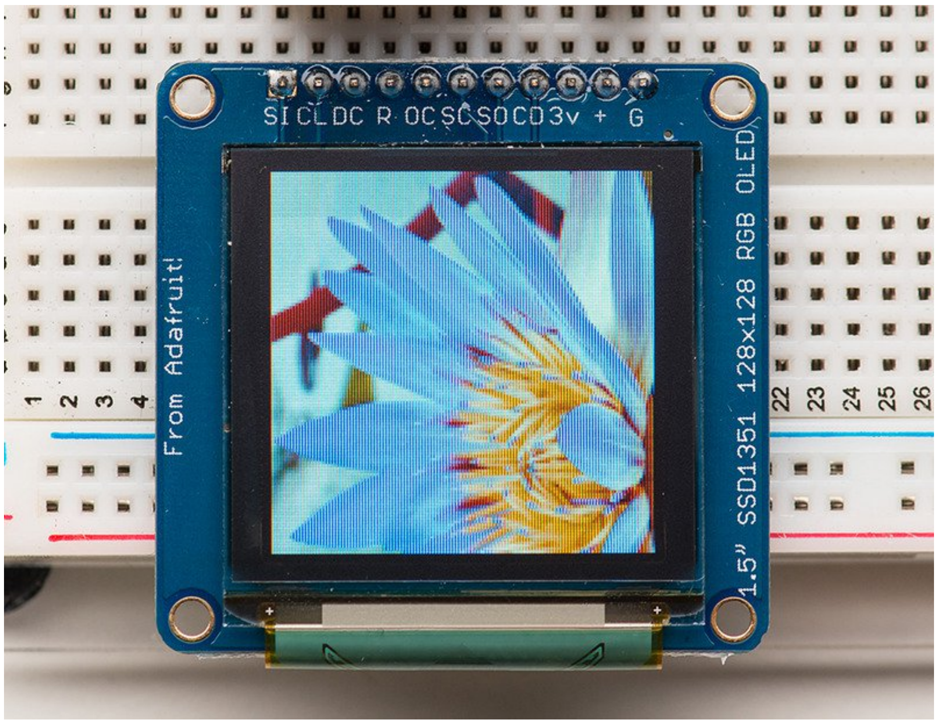
Last updated on 2019-04-16 06:43:28 AM UTC

Overview



We love our black and white monochrome displays but we also like to dabble with some color now and then. Our big 1.5" color OLED displays are perfect when you need a small display with vivid, high-contrast 16-bit color. The visible portion of the OLED measures 1.5" diagonal and contains 128x128 RGB pixels, each one made of red, green and blue OLEDs. Each pixel can be set with 16-bits of resolution for a large range of colors. Because the display uses OLEDs, there is no backlight, and the contrast is very high (black is really black). We picked this display for its excellent color, this is the nicest mini OLED we could find!

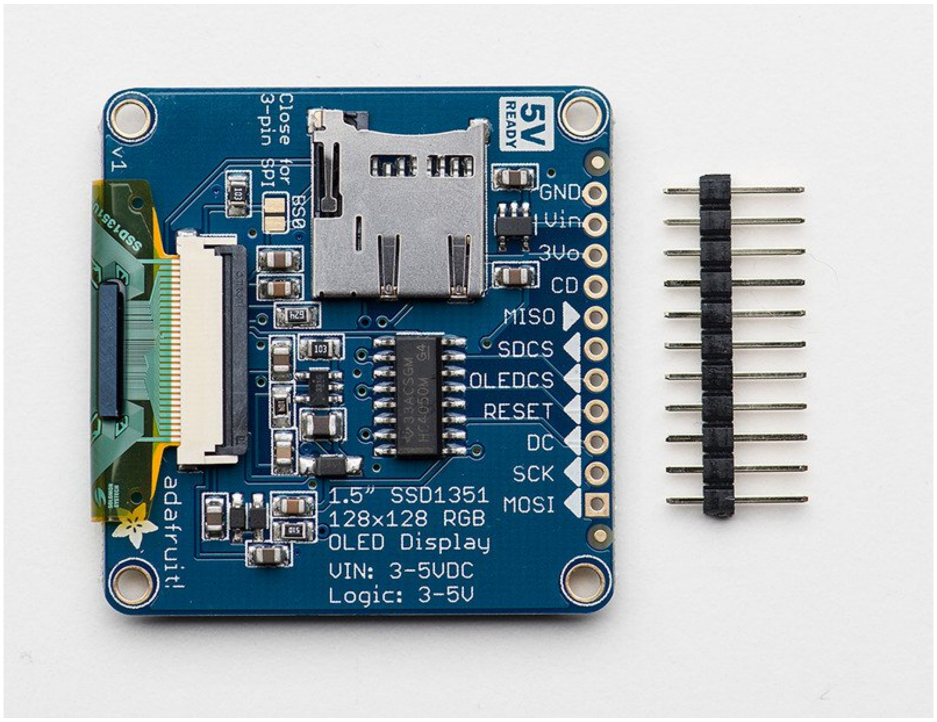
This OLED uses the SSD1351 driver chip, which manages the display. You can talk to the driver chip using 4-wire write-only SPI (clock, data, chip select, data/command and an optional reset pin). Included on the fully assembled breakout is the OLED display and a small boost converter (required for providing 12V to the OLED) and a microSD card holder. This design includes built-in logic level shifting so you can use it with 3-5VDC power and logic levels. Our example code shows how to read a bitmap from the uSD card and display it all via SPI.

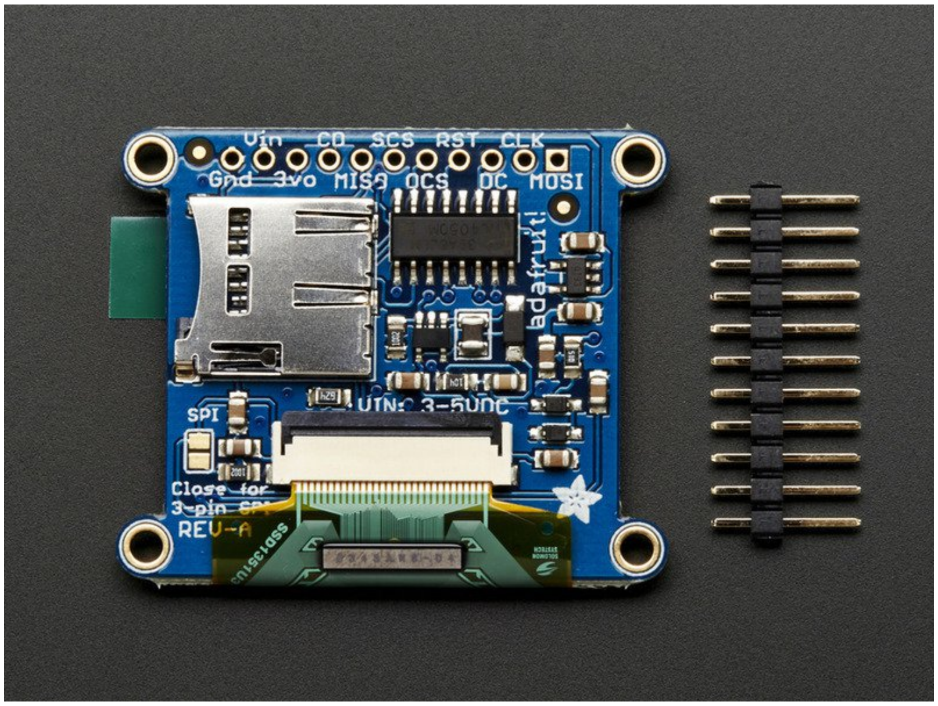


Board Technical Details

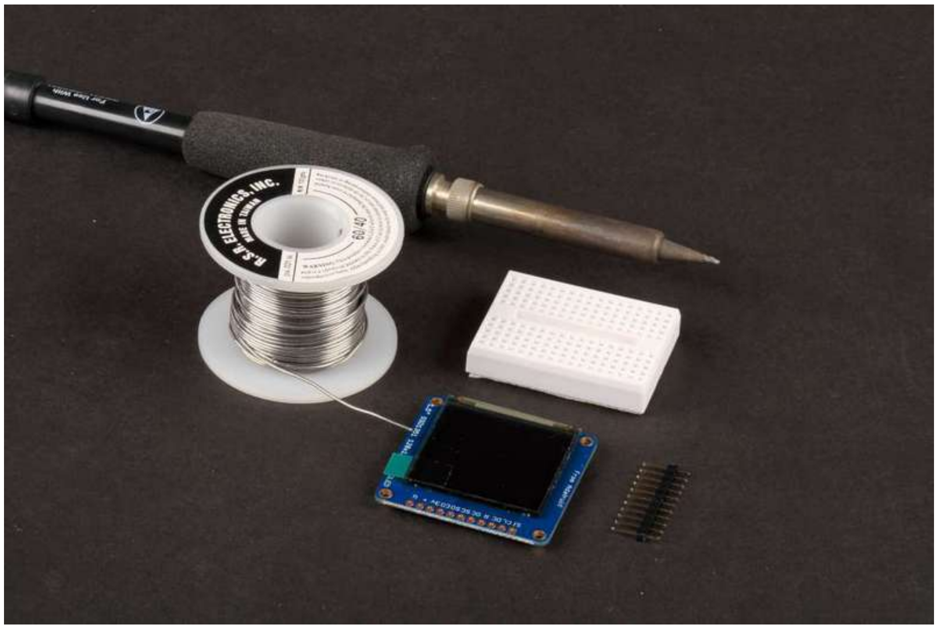
- 1.5" diagonal OLED, 16-bit color
- SPI interface
- 3.3-5V logic and power
- Micro-SD card holder
- Dimensions: 43.17mm / 1.7" x 42mm / 1.65" x 5.42mm / 0.2"

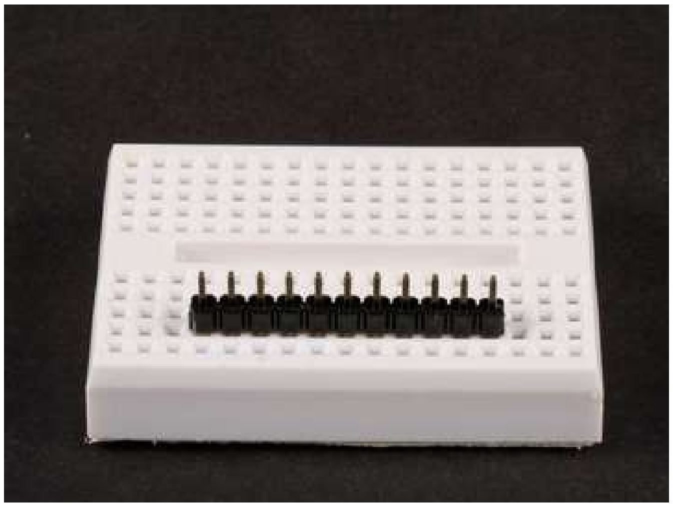
Assembly





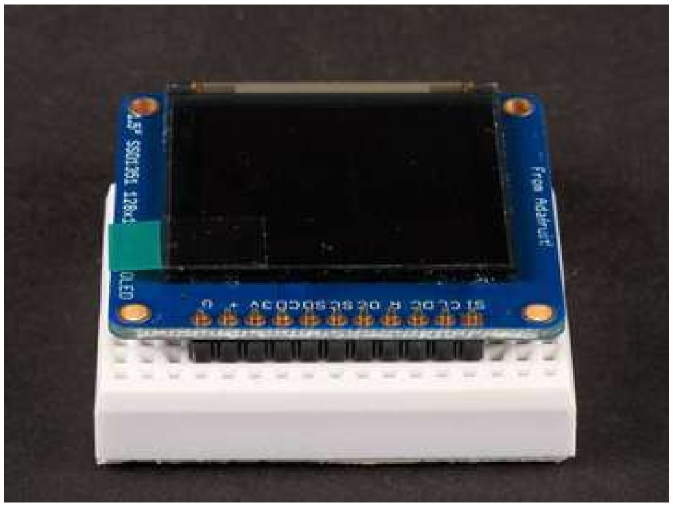
The breakout board comes fully assembled and tested. We include an optional strip of header pins to make it easier to use this display in a breadboard. The header can be installed in just a few minutes with your soldering iron:





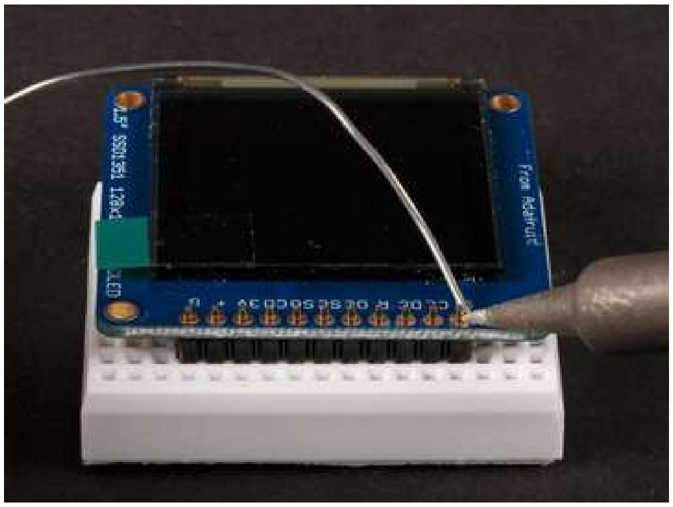
Prepare the header strip

Cut the header to size and insert (long pins down) into a breadboard to stabilize for soldering.



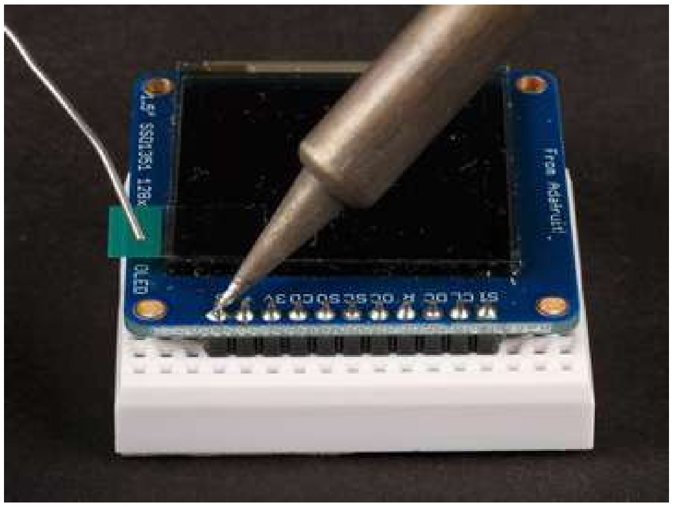
Position the display

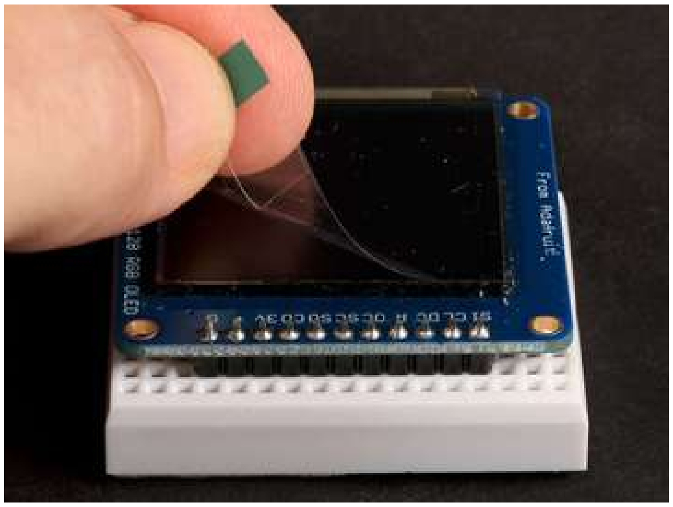
Place the display breakout on the header so that the short pins protrude through the holes.



And Solder!

Solder all pins to assure a good electrical connection.



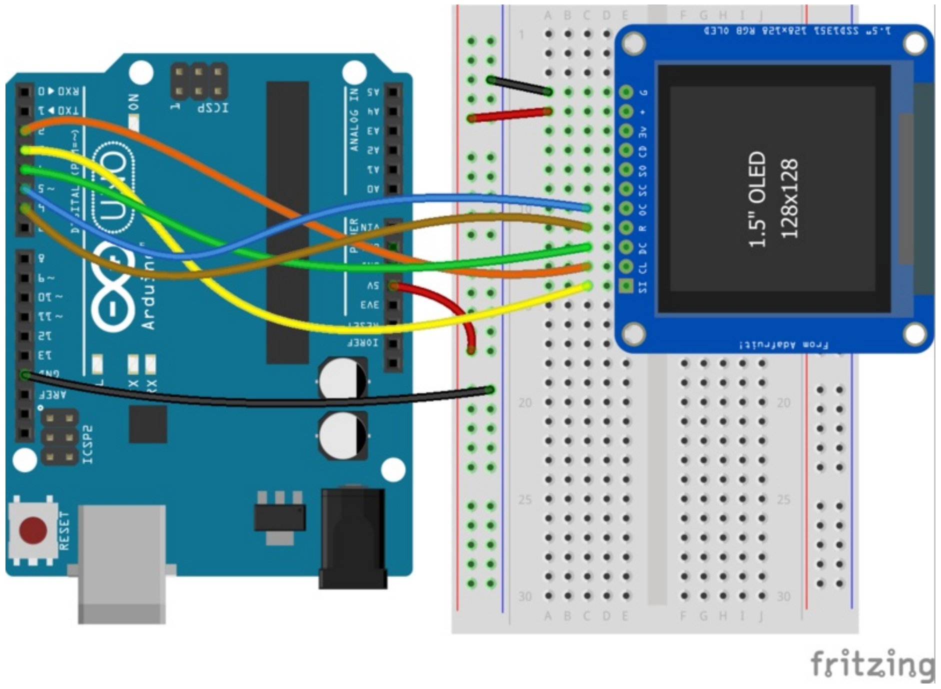


Remove the protective film

Gently pull up on the tab to remove the film.

Wiring and Graphics Test

The pinout ordering is the same for both the 1.27" and 1.5" version of the OLED!



https://adafru.it/sVa

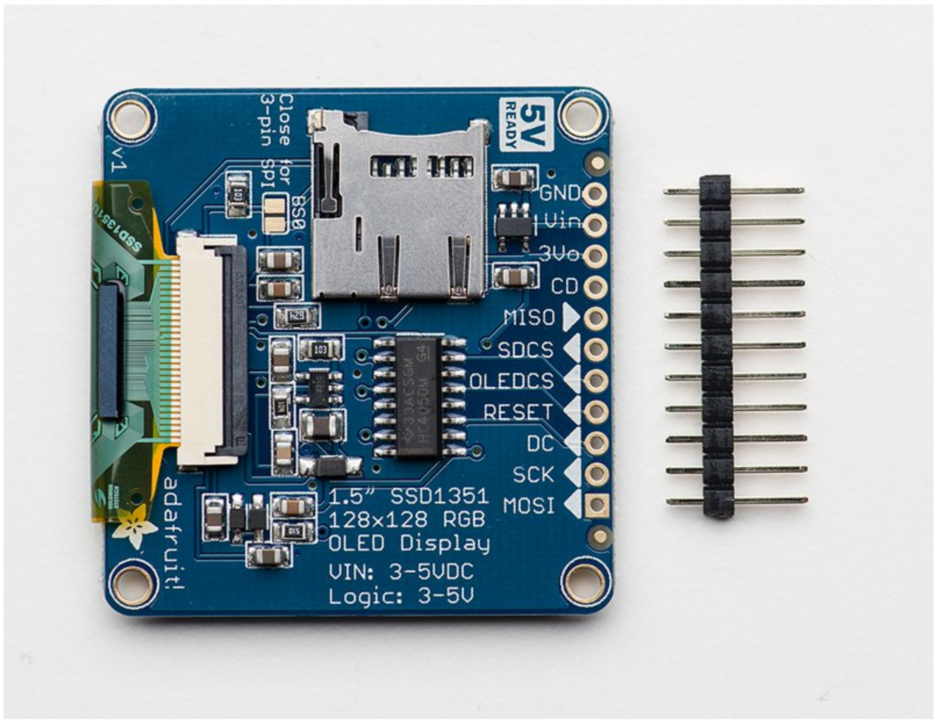
https://adafru.it/sVa

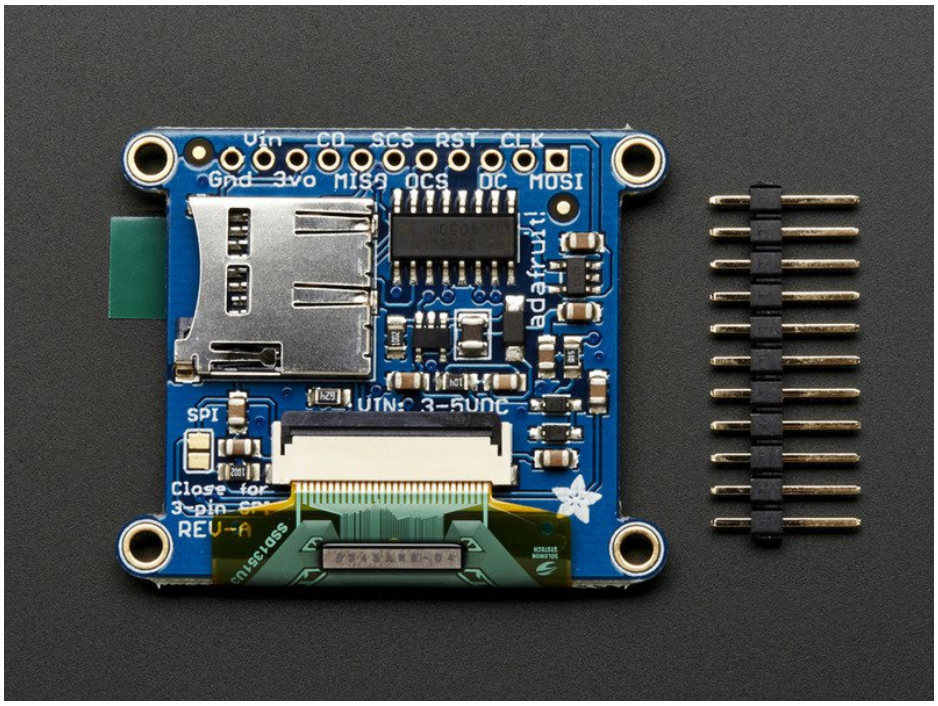
The library supports flexible wiring to minimize pin conflicts with other shields and breakouts. For the initial test, we'll use the same wiring as the "test" example from the library:

- **GND -> GND (G)**
- **5v -> VIN (+)**
- **#2 -> SCLK (CL)**
- **#3 -> MOSI (SI)**
- **#4 -> DC**
- **#5 -> OLEDCS (OC)**
- **#6 -> RST (R)**

Hint:

If you are confused by the abbreviations on the front of the board, the full signal names are printed on the back!

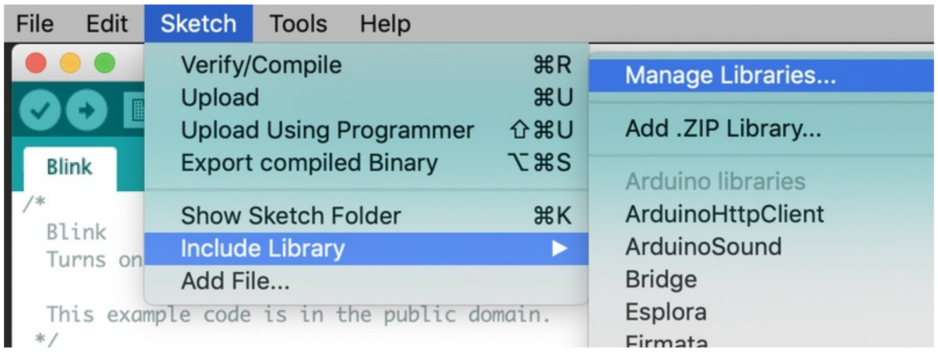




Installing the Arduino software

Now we can run the test software on the Arduino.

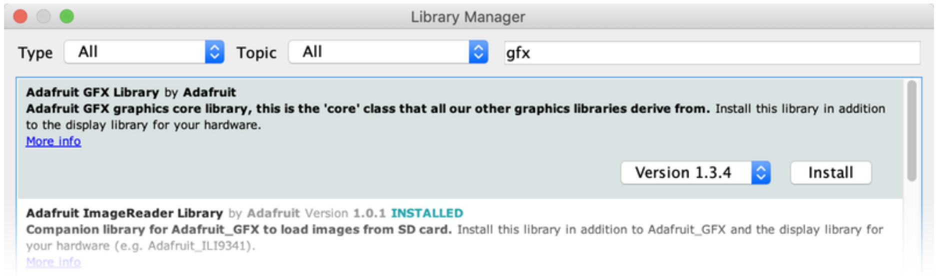
Three libraries need to be installed using the **Arduino Library Manager**…this is the preferred and modern way. From the Arduino “Sketch” menu, select “Include Library” then “Manage Libraries…”



```
/*
  Blink
  Turns on

  This example code is in the public domain.
 */
```

Type “gfx” in the search field to quickly find the first library — **Adafruit_GFX**:



Adafruit GFX Library by **Adafruit**
Adafruit GFX graphics core library, this is the 'core' class that all our other graphics libraries derive from. Install this library in addition to the display library for your hardware.

Adafruit ImageReader Library by Adafruit Version **1.0.1** **INSTALLED**
Companion library for Adafruit_GFX to load images from SD card. Install this library in addition to Adafruit_GFX and the display library for your hardware (e.g. Adafruit_ILI9341).

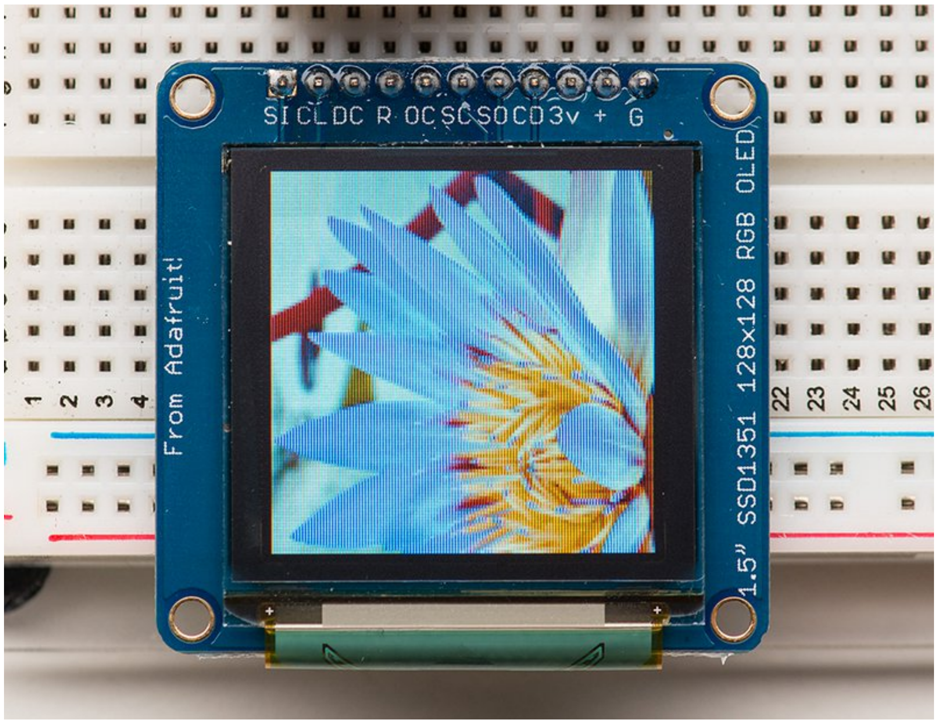
Repeat the search and install steps, looking for the **Adafruit_ZeroDMA** and **Adafruit_SSD1351** libraries.

After you restart, you should be able to select **File→Examples→Adafruit_SSD1351→test** - this is the example sketch that just tests the display by drawing text and shapes. Upload the sketch and you should see the following:

The test sketch demonstrates all the basic drawing functions of the Adafruit GFX Library. Read through the code to see how to draw text, circles, lines, etc.

For a detailed tutorial on the Adafruit GFX library, including all the functions available please visit the GFX tutorial page (https://adafru.it/aPx)

Drawing Bitmaps

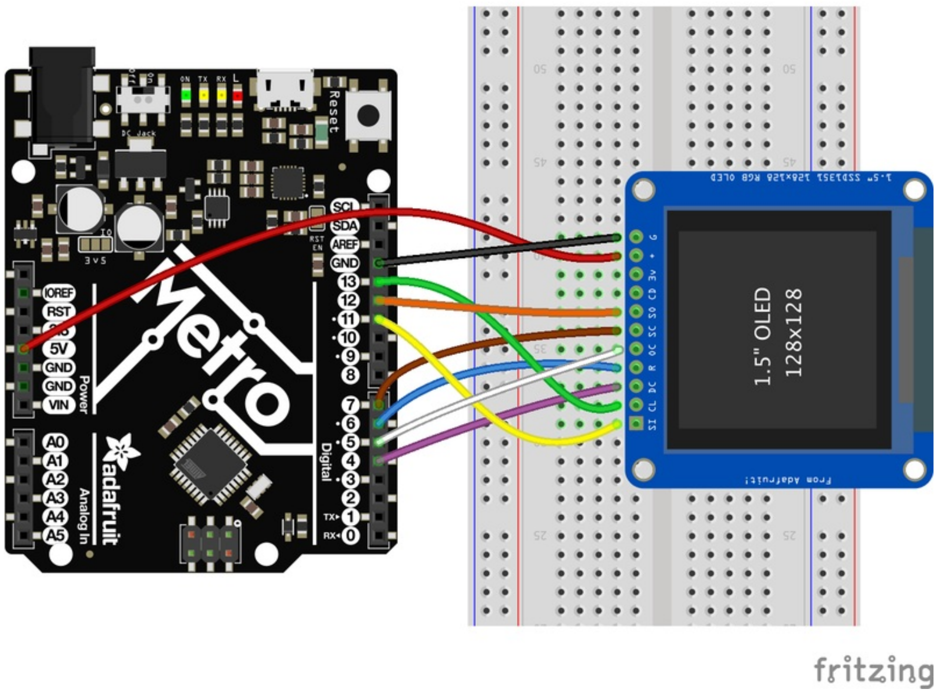


Wiring for the Bitmap Example

Drawing bitmaps from the on-board micro SD card requires a few more connections to communicate with the SD card. The library allows you to use any pins. The Arduino connections listed below match the code in the "bmp" example from the library:

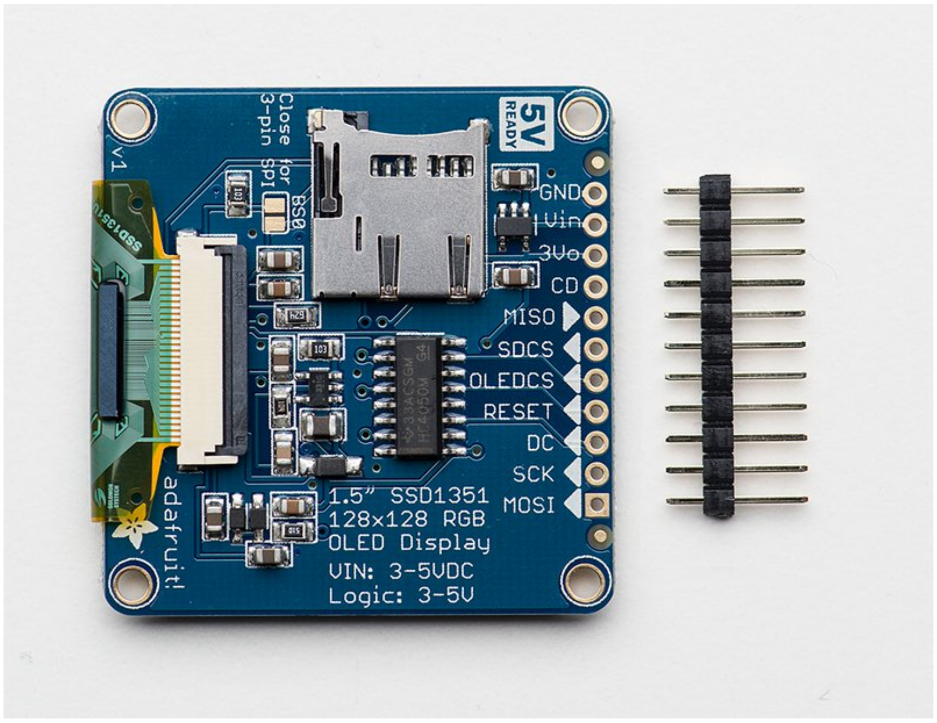
- **GND -> GND (G)**
- **5v -> VIN (+)**
- **#7 -> SDCS (SC)**
- **#4 -> DC**
- **#6 -> RST (R)**
- **#5 -> OLEDCS (OC)**
- **#11 -> MOSI (SI)**
- **#12 -> MISO (SO)**
- **#13 -> SCLK (CL)**

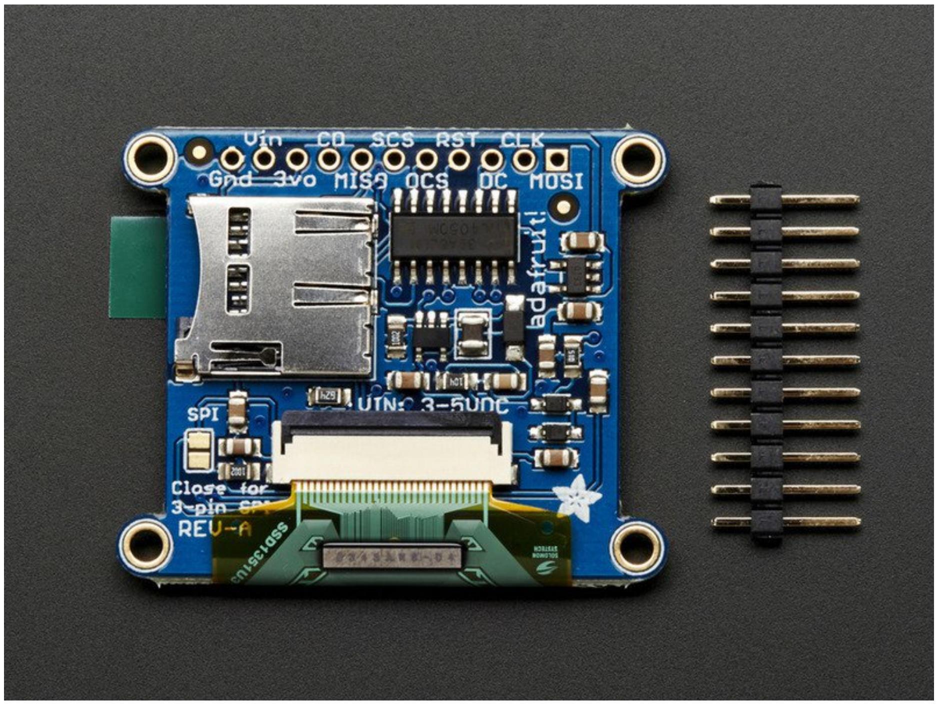
Note that the Bitmap example code uses hardware SPI wiring for maximum speed. You can still use software SPI, but make sure that the pin definitions match your wiring and that you modify the example to select the Software SPI option (#1) in the code. The SPI pins shown are for Atmega-328 processors. To use this wiring on other processors, software SPI must be used.



Hint:

If you are confused by the abbreviations on the front of the board, the full signal names are printed on the back!



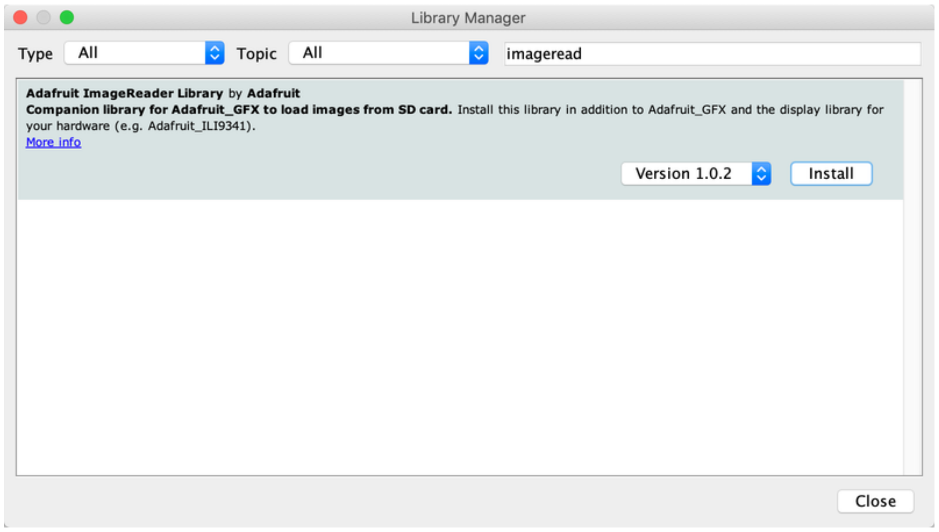


Bitmap Example Sketch

To display bitmaps from the on-board micro SD slot, you will need a micro SD card (http://adafru.it/102) formatted **FAT16 or FAT32** (they almost always are by default).

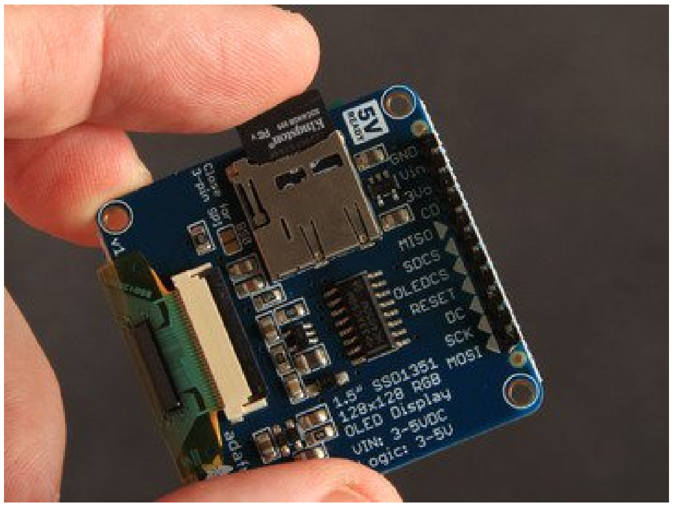
There is a built in microSD card slot on the rear of the breakout and we can use that to load bitmap images!

It's really easy to draw bitmaps. We have a library for it, Adafruit_ImageReader_Library, which can be installed through the Arduino Library Manager (Sketch→Include Library→Manage Libraries…). Enter “imageread” in the search field and the library is easy to spot:



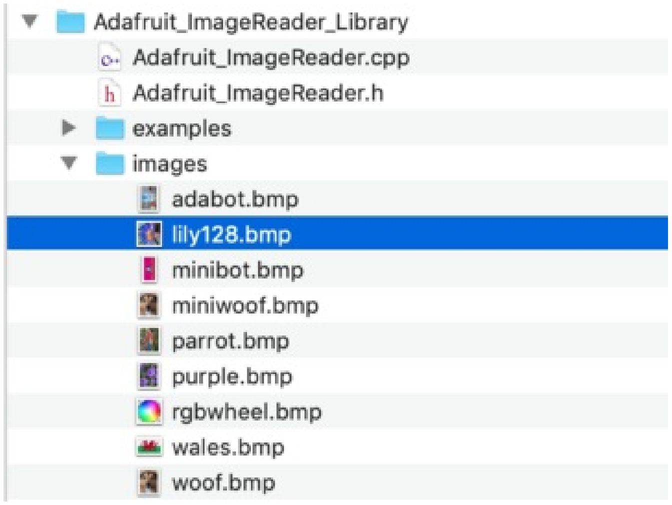
imageread

Adafruit ImageReader Library by **Adafruit**
Companion library for Adafruit_GFX to load images from SD card. Install this library in addition to Adafruit_GFX and the display library for your hardware (e.g. Adafruit_ILI9341).
More info



Insert the card

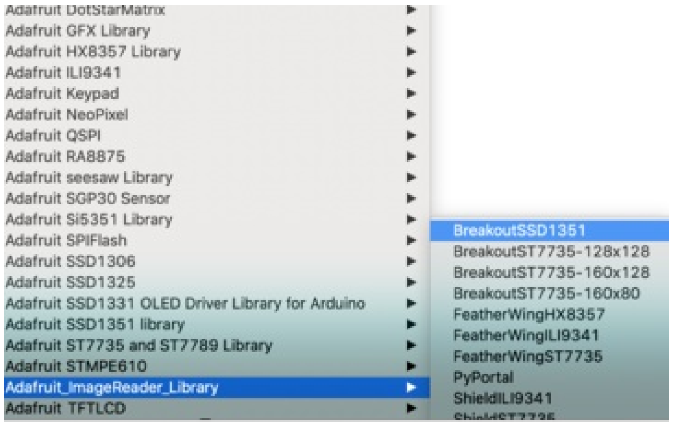
Insert the micro SD card into the slot on the back of the SSD1351 breakout board.



```
Adafruit_ImageReader_Library
    Adafruit_ImageReader.cpp
    Adafruit_ImageReader.h
    examples
    images
        adabot.bmp
        lily128.bmp
        minibot.bmp
        miniwoof.bmp
        parrot.bmp
        purple.bmp
        rgbwheel.bmp
        wales.bmp
        woof.bmp
```

Copy the bitmap file

Copy the file **"lily128.bmp"** from the Adafruit_ImageReader_Library\images folder to the root directory of your micro-SD card.



Load the BreakoutSSD1351 example sketch

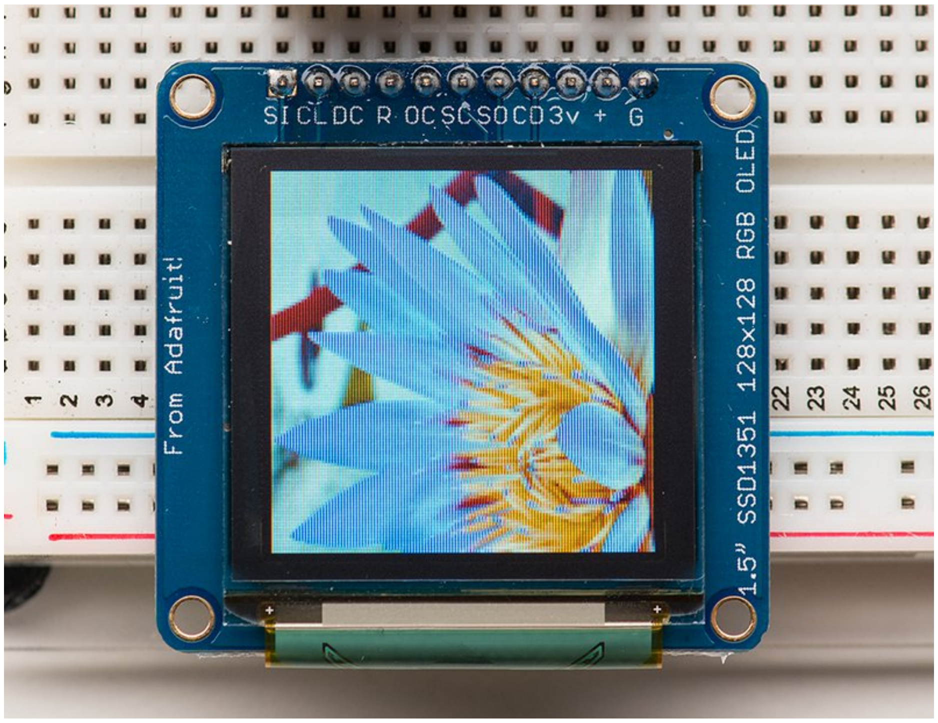
Select **"Examples->Adafruit_ImageReader_Library->BreakoutSSD1351"** to load it into your editor.

In the example, find the following section of code:

```
// Load full-screen BMP file 'rgbwheel.bmp' at position (0,0) (top left).
// Notice the 'reader' object performs this, with 'tft' as an argument.
Serial.print(F("Loading rgbwheel.bmp to screen..."));
stat = reader.drawBMP("/rgbwheel.bmp", tft, 0, 0);
reader.printStatus(stat);   // How'd we do?
```

On the line with `reader.drawBMP()` change `"/rgbwheel.bmp"` to `"/lily128.bmp"`.

After that, upload it to your Arduino and when the Arduino restarts, you should see the flower as below!



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 to decode. You can rotate images using the **setRotation()** procedure.

The BreakoutSSD1351 example sketch shows everything you need to work with BMP images. Here's just the vital bits broken out...

Several header files are included at the top of the sketch. All of these are required...they let us access the SD card and the display, and provide the image-reading functions:

```
#include <SPI.h>
#include <SD.h>
#include <Adafruit_GFX.h>         // Core graphics library
#include <Adafruit_SSD1351.h>     // Hardware-specific library
#include <Adafruit_ImageReader.h> // Image-reading functions
```

Several #defines relate to hardware pin numbers, all fixed values when using the shield.

Then we declare the tft screen object, and the image-reader object like so:

```
#define SD_CS    7 // SD card select pin
#define TFT_CS   5 // TFT select pin
#define TFT_DC   4 // TFT display/command pin
#define TFT_RST  6 // Or set to -1 and connect to Arduino RESET pin

Adafruit_SSD1351 tft = Adafruit_SSD1351(SCREEN_WIDTH, SCREEN_HEIGHT, &SPI, TFT_CS, TFT_DC, TFT_RST);

Adafruit_ImageReader reader;     // Class w/image-reading functions
```

After the SD and TFT's `begin()` functions have been called (see the example sketch again, in the `setup()` function), you can then call `reader.drawBMP()` to load an image from the card to the screen:

```
ImageReturnCode stat; // Status from image-reading functions
stat = reader.drawBMP("/lily128.bmp", tft, 0, 0);
```

You can draw as many images as you want — though remember the names must be less than 8 characters long. Call like so:

```
reader.drawBMP(filename, tft, x, y);
```

'x' and 'y' are pixel coordinates where top-left corner of the image will be placed. Images can be placed anywhere on screen...even partially off screen, the library will clip the section to load.

Image loading is explained in greater depth in the Adafruit_GFX library guide. (https://adafru.it/DpM)

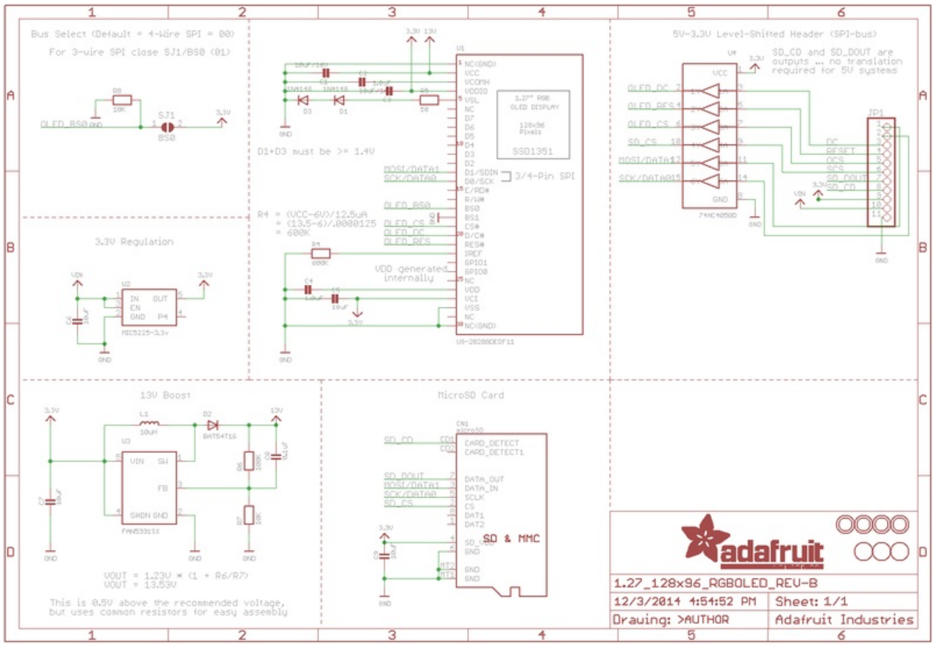
Downloads and Links

Data Sheets:

- SSD1351 Display Controller Datasheet (https://adafru.it/sVb)
- 1.5" OLED Display Module datasheet (https://adafru.it/cBE)
- Fritzing objects in the Adafruit Fritzing library (https://adafru.it/aP3)
- EagleCAD PCB for 1.27" Color OLED (https://adafru.it/rqB)
- EagleCAD PCB for the 1.5" Color OLED (https://adafru.it/rqC)

Schematic

Click to enlarge


Bus Select (Default = 4-Wire SPI = 00)
For 3-wire SPI close SJ1/BS0 (01)
3.3V Regulation
13V Boost
MicroSD Card
5V-3.3V Level-Shifted Header (SPI-bus)
SD_CD and SD_DOUT are outputs .. no translation required for 5V systems
SSD1351
D1+D3 must be >= 1.4V
VOUT = 1.23V * (1 + R6/R7)
VOUT = 13.53V
This is 0.5V above the recommended voltage, but uses common resistors for easy assembly
SD & MMC
1.27_128x96_RGBOLED_REV-B
12/3/2014 4:54:52 PM
Sheet: 1/1
Drawing: >AUTHOR
Adafruit Industries

For the level shifter we use the CD74HC4050 (https://adafru.it/Boj) which has a typical propagation delay of ~10ns