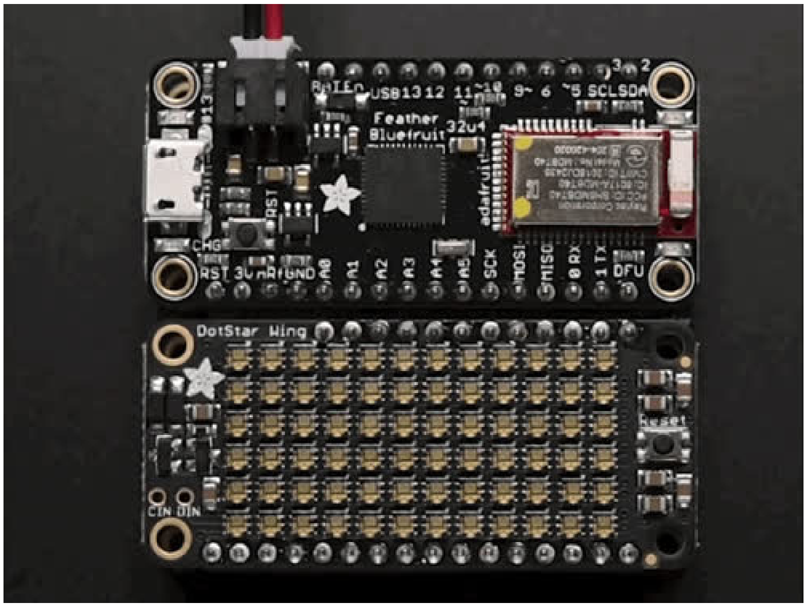
adafruit learning system

Adafruit DotStar FeatherWing

Created by lady ada

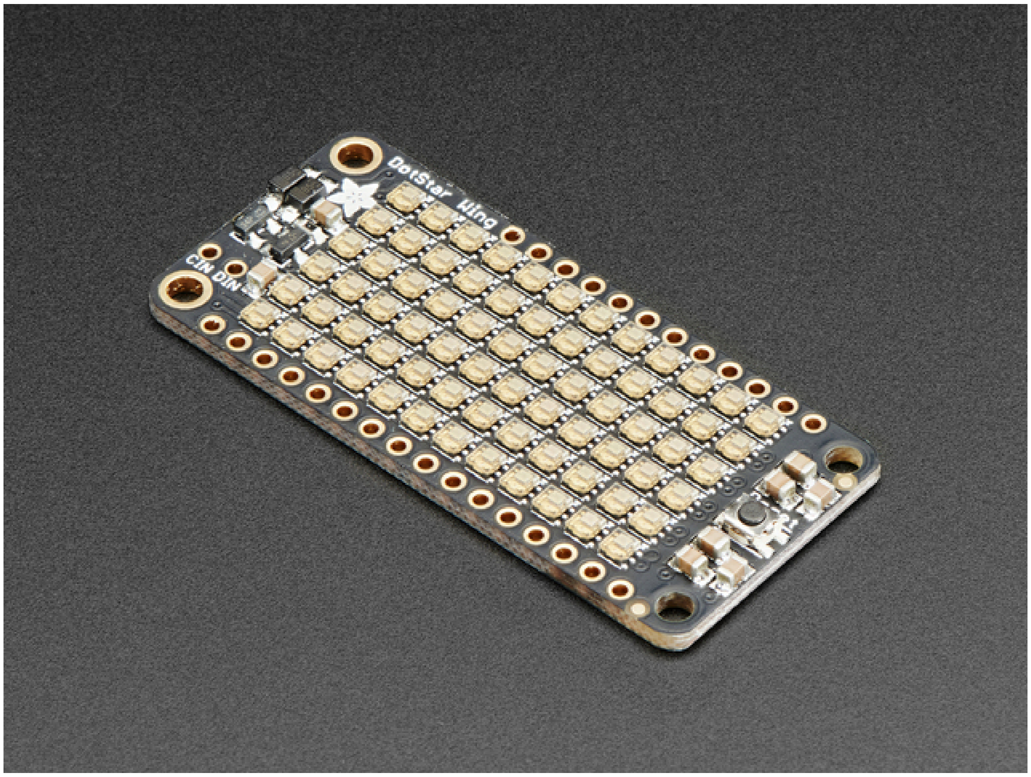


Last updated on 2018-08-22 04:03:05 PM UTC

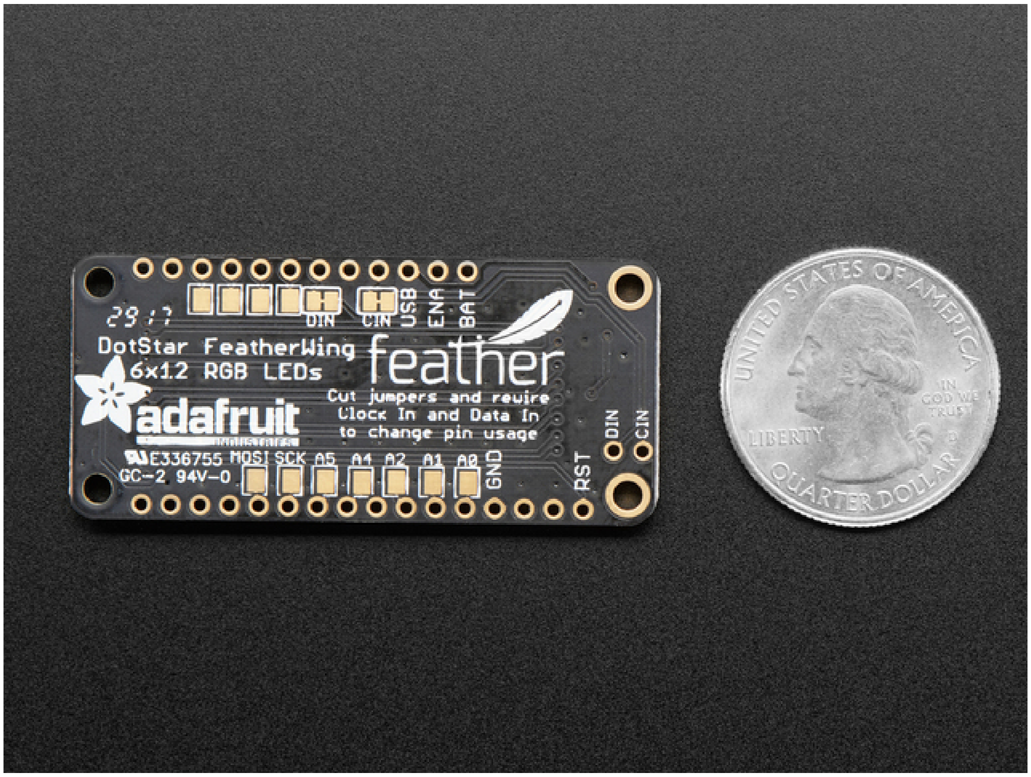
Guide Contents

Overview

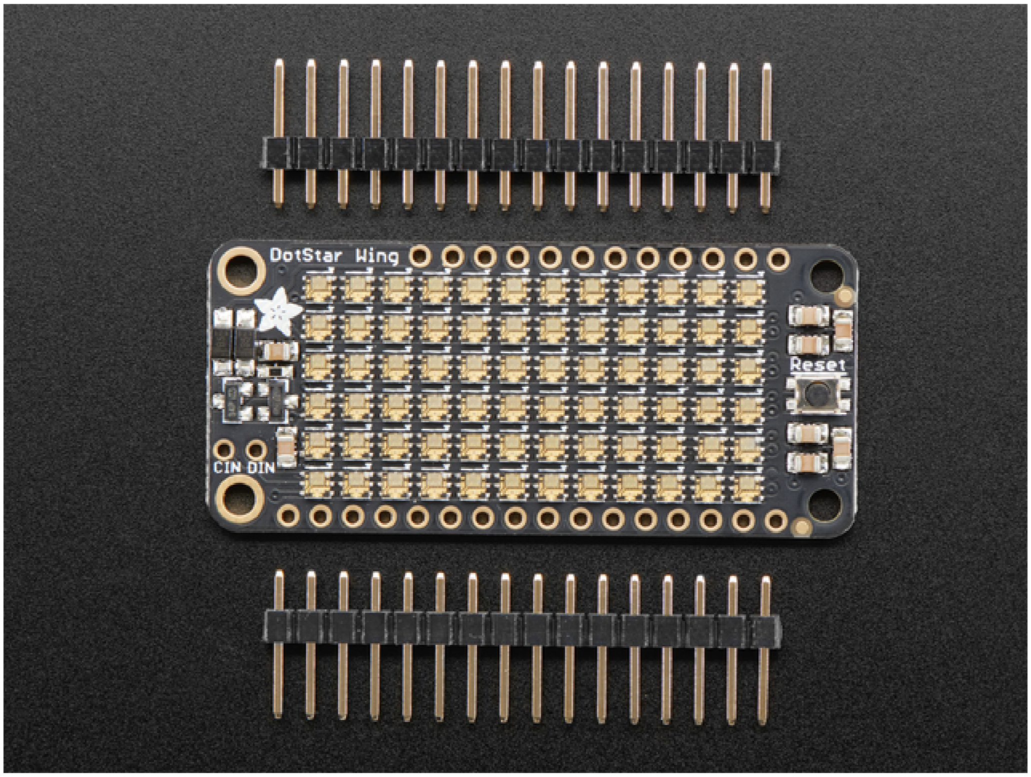
A Feather board without ambition is a Feather board without FeatherWings! This is the **DotStar FeatherWing, a 6x12 RGB LED Add-on For All Feather Boards**! Using our Feather Stacking Headers (http://adafru.it/2830) or Feather Female Headers (http://adafru.it/2886) you can connect a FeatherWing on top or bottom of your Feather board and make your Feather board strut like a peacock at a rave.



Put on your sunglasses before staring into these 72 configurable RGB LEDs, they are super bright! Arranged in a 6x12 matrix, each 2mm by 2mm sized RGB pixel is individually addressable. Only two pins are required to control all the LEDs. On the bottom we have jumpers for the Data and Clock lines so you can change them from the defaults. **Works with any/all of our Feathers!** You can cut the default jumper traces and use any pins you like

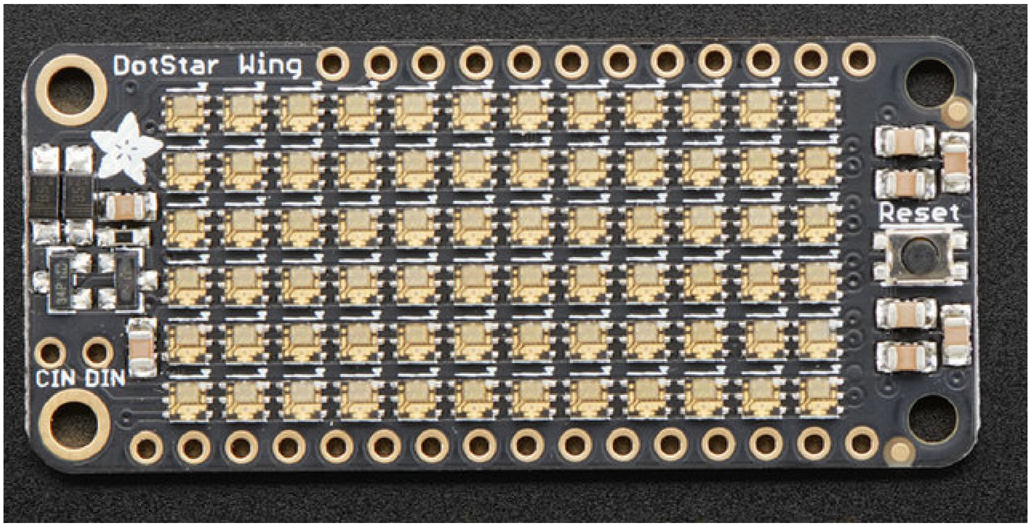


To make it easy to start, the LEDs are by default powered from *either* the USB power line or Battery power line, whichever is higher. Two Schottky diodes are used to switch between the two. This power arrangement is able to handle 1 Amp of constant current draw and maybe 2A peak, so not a good way to make a flashlight. It's better for colorful effects. A power MOSFET is set up and connected to the ENABLE pin on the Feather so that when the Feather is disabled, the wing will also cut power.



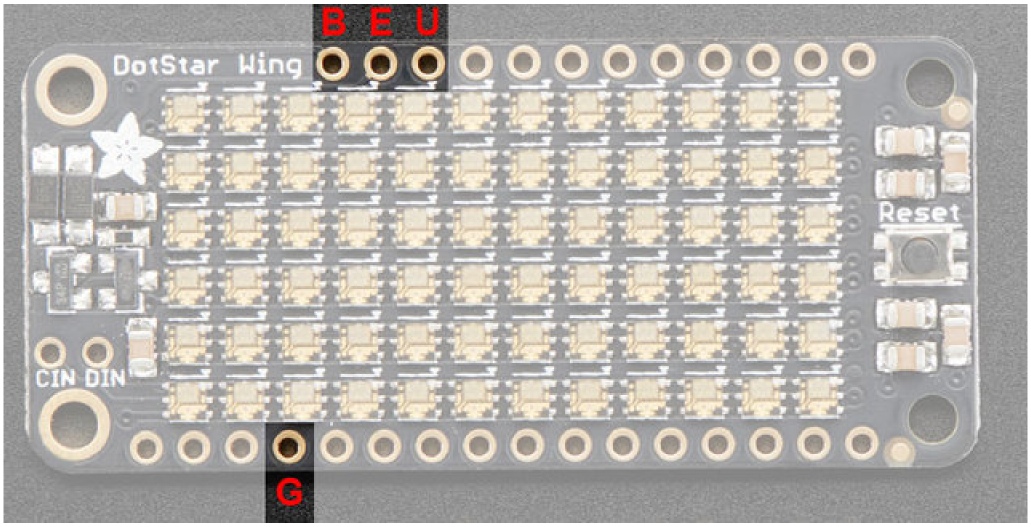
Pinouts

The DotStar FeatherWing is pretty easy to use, there's only one data pin and one clock pin.



Power Pins

There are four power-related pins:



The **B**attery and **U**sb power pins are used to power the many DotStars. There's a lot of pixels and you can't really run them from the 3.3V regulator. Two 1A Schottkey diodes will automatically select whichever voltage is higher. That means, if USB is plugged in, you'll get ~5V USB power. If the feather is running off of battery power, say a LiPoly battery, you'll get 3.7-4.2V power. While this is a little lower than spec, DotStars are only a little dimmer.

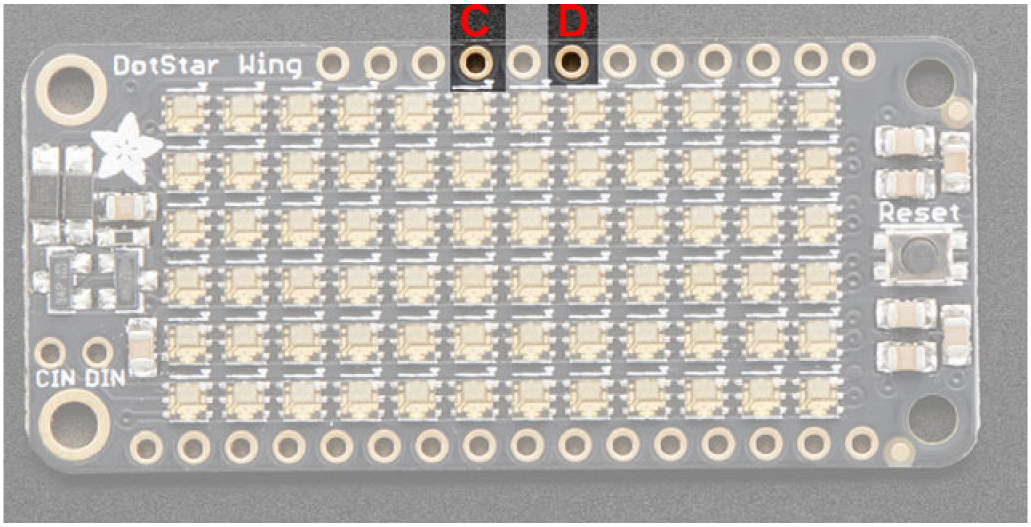
The **E**nable pin is tied to an inverter + MOSFET and will cut power to the matrix when low. By default, all Feather's have this pin pulled high

The **G**round pin is common data/power ground.

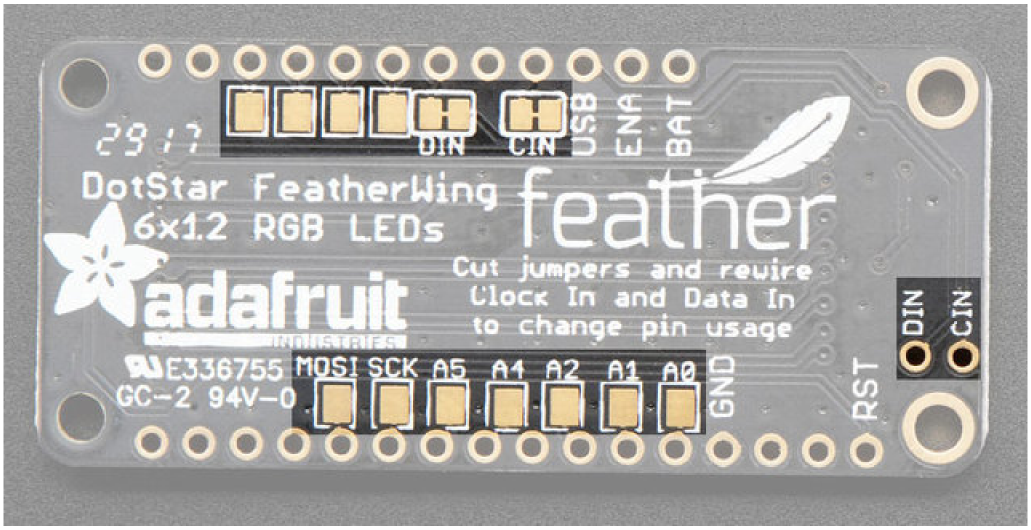
Data Pins

Unlike NeoPixels, DotStar LEDs require *two* pins - a data and clock. While it's best if these are on a hardware SPI port for speed, you can use any two pins with Arduino or CircuitPython.

By default, these are the **D**ata and **C**lock pins:

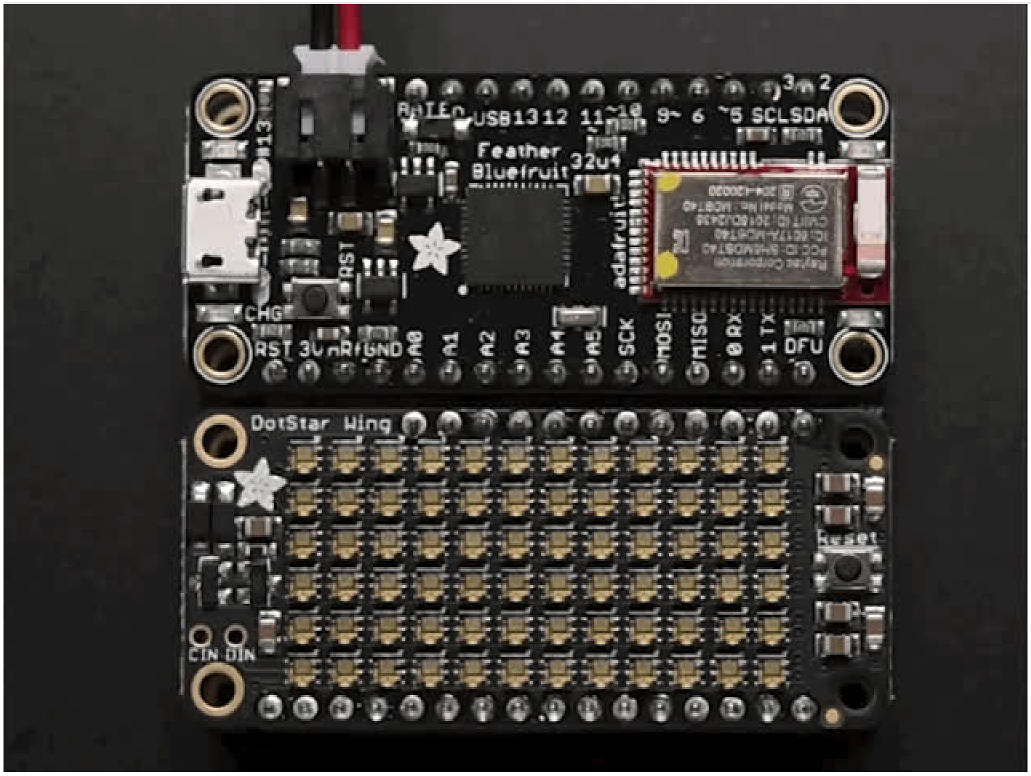


But - it's very easy to change the connections. On the bottom, you can cut the two jumper traces near those two pins, then solder wires from the **CIN** and **DIN** pins to the available square pads



Note there is no logic level shifter for Clock and Data - we have not found that these chips require a level shifter, they work nicely at 3.3V logic even with 5V+ power

Usage



Once you've soldered headers on, check out https://learn.adafruit.com/adafruit-dotstar-leds/ (https://adafru.it/zzf)for everything you could possibly ever want to know about DotStars. It's a handy guide for timing information, power calculations, etc.

DotMatrix Usage

Here's a demo to get you started. Once you've installed the Adafruit DotStar and DotStar_Matrix libraries (you can get more info on how to that here (https://adafru.it/zzf)) This example, when loaded onto your Feather will give you the pretty ADAFRUIT! scrolling demo. We've added a 5-pixel tall font so you can easily make scrolling messages!

```
// Adafruit_DotStarMatrix example for single DotStar LED matrix.
// Scrolls 'Adafruit' across the matrix.

#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_DotStarMatrix.h>
#include <Adafruit_DotStar.h>
#include <Fonts/TomThumb.h>

#if defined(ESP8266)
  #define DATAPIN    13
  #define CLOCKPIN   14
#elif defined(__AVR_ATmega328P__)
  #define DATAPIN    2
  #define CLOCKPIN   4
#elif defined(ARDUINO_SAMD_FEATHER_M0)
  #define DATAPIN    11
  #define CLOCKPIN   13
#elif defined(ARDUINO_AVR_FEATHER32U4)
  #define DATAPIN    11
  #define DATAPIN    11
  #define CLOCKPIN   13
#elif defined(ARDUINO_FEATHER52)
  #define DATAPIN    7
  #define CLOCKPIN   16
#elif defined(TEENSYDUINO)
  #define DATAPIN    9
  #define CLOCKPIN   5
#elif defined(ARDUINO_ARCH_WICED)
  #define DATAPIN    PA4
  #define CLOCKPIN   PB5
#elif defined(ESP32)
  #define DATAPIN    27
  #define CLOCKPIN   13
#endif

#define SHIFTDELAY 100
#define BRIGHTNESS 20

// MATRIX DECLARATION:
// Parameter 1 = width of DotStar matrix
// Parameter 2 = height of matrix
// Parameter 3 = pin number (most are valid)
// Parameter 4 = matrix layout flags, add together as needed:
//   DS_MATRIX_TOP, DS_MATRIX_BOTTOM, DS_MATRIX_LEFT, DS_MATRIX_RIGHT:
//     Position of the FIRST LED in the matrix; pick two, e.g.
//     DS_MATRIX_TOP + DS_MATRIX_LEFT for the top-left corner.
//   DS_MATRIX_ROWS, DS_MATRIX_COLUMNS: LEDs are arranged in horizontal
//     rows or in vertical columns, respectively; pick one or the other.
//   DS_MATRIX_PROGRESSIVE, DS_MATRIX_ZIGZAG: all rows/columns proceed
//     in the same order, or alternate lines reverse direction; pick one.
//   See example below for these values in action.
// Parameter 5 = pixel type:
//   DOTSTAR_BRG  Pixels are wired for BRG bitstream (most DotStar items)
//   DOTSTAR_GBR  Pixels are wired for GBR bitstream (some older DotStars)
//   DOTSTAR_BGR  Pixels are wired for BGR bitstream (APA102-2020 DotStars)

Adafruit_DotStarMatrix matrix = Adafruit_DotStarMatrix(
                                  12, 6, DATAPIN, CLOCKPIN,
                                  DS_MATRIX_BOTTOM     + DS_MATRIX_LEFT +
                                  DS_MATRIX_ROWS + DS_MATRIX_PROGRESSIVE,
                                  DOTSTAR_BGR);

const uint16_t primaryColors[] = {
  matrix.Color(255, 0, 0), matrix.Color(0, 255, 0), matrix.Color(0, 0, 255)
};

const uint16_t adaColors[] = {
  matrix.Color(255, 0, 0),   //A red
  matrix.Color(255, 125, 0), //D orange
  matrix.Color(200, 255, 0), //A yellowish
  matrix.Color(0, 255, 0),   //F green
  matrix.Color(0, 255, 225), //R blue
  matrix.Color(150, 0, 255), //U purple
  matrix.Color(255, 0, 220), //I pink
  matrix.Color(255, 65, 0),  //T reddish
  matrix.Color(255, 220, 0)  //! orange/yellow
};

char adafruit[] = "ADAFRUIT!";
```

```
void setup() {
  Serial.begin(115200);

  // uncomment to have wait
  //while (!Serial) delay(500);

  Serial.println("\nDotstar Matrix Wing");
  matrix.begin();
  matrix.setFont(&TomThumb);
  matrix.setTextWrap(false);
  matrix.setBrightness(BRIGHTNESS);

  for (byte i = 0; i < 3; i++) {
    matrix.fillScreen(primaryColors[i]);
    matrix.show();
    delay(500);
  }
}

int x = matrix.width();
int pass = 0;

void loop() {
  matrix.fillScreen(0);
  matrix.setCursor(x, 5);
  for (byte i = 0; i < 9; i++) {
    // set the color
    matrix.setTextColor(adaColors[i]);
    // write the letter
    matrix.print(adafruit[i]);
  }

  if (--x < -50) {
    x = matrix.width();
  }

  matrix.show();
  delay(SHIFTDELAY);
}
```

With the DotStarMatrix library, we're using Adafruit_GFX, our graphics underpinning library. That means you can draw lines, dots, text, triangles, rectangles, bitmaps, what-have-you very easily across all Adafruit displays.

For more details on Adafruit GFX, check our the learning guide at https://learn.adafruit.com/adafruit-gfx-graphics-library/ (https://adafru.it/zra)

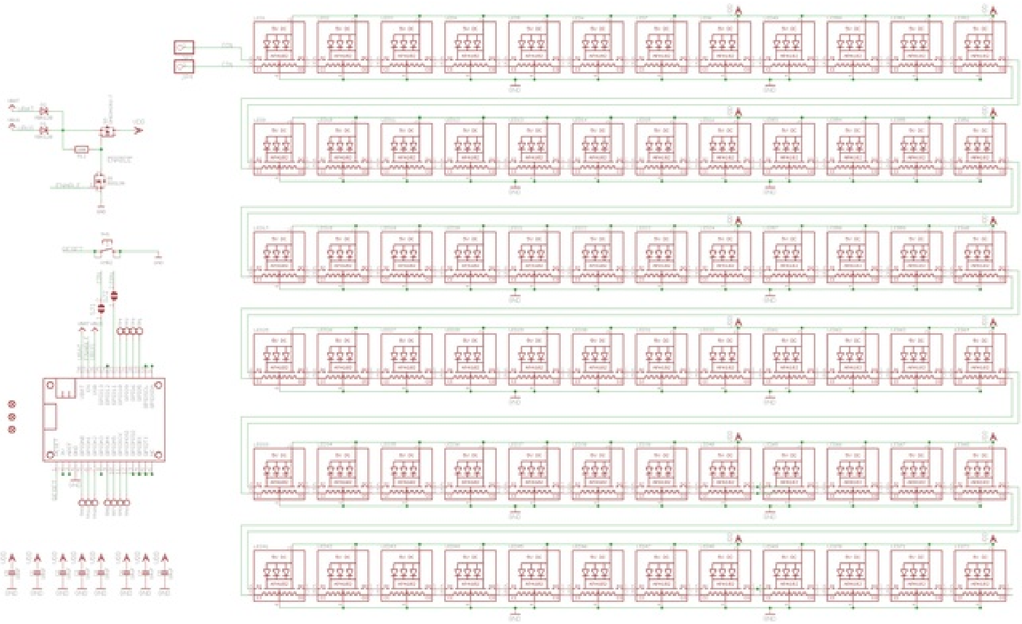
Downloads

Files

- EagleCAD PCB Files on GitHub (https://adafru.it/zzA)
- Fritzing object available in the Adafruit Fritzing Library (https://adafru.it/aP3)

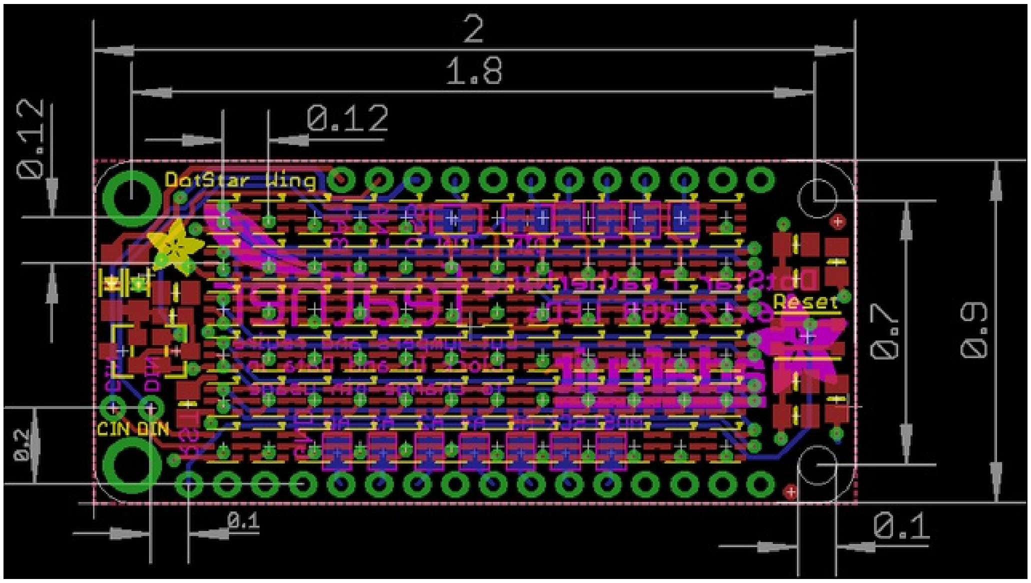
Schematic

Click to embiggen



Fabrication Print

Dimensions in inches


2
1.8
0.12
0.12
DotStar Wing
Reset
CIN DIN
0.7
0.9
0.2
0.1
0.1