

ARDUINO

Controlling WS2812B Addressable LEDs with Arduino





Whether you call them individually addressable RGB LEDs, WS2812B, or NeoPixels, there's no denying that they are extremely popular and a must-have for any glowy and blinky project.

What sets these LEDs apart is their unique ~~feature~~ feature that allows you to individually address and control the color and brightness of each LED on a strip. Plus, you can (theoretically) chain as

Controlling WS2812B Addressable LEDs with Arduino

many LEDs as you want and control them all using a single microcontroller pin. The more LEDs you chain together, the more fancier your animations will be!

These LEDs are exceptionally bright, produce vibrant colors, and are easy to hookup. This tutorial explains everything you need to know about WS2812B addressable LEDs and how to use them with an Arduino.



Table Of Contents

- Introduction
- WS2812B Hardware Overview
 - Color-Depth
 - Chain-able
 - Power
- WS2812B Data Transfer Protocol
- WS2812B Strip Pinout
 - The Input Connector
 - The Output Connector
 - The Auxiliary Power Wires

Whether you call them individual addressable RGB LEDs, WS2812B, or NeoPixels, there's no denying that they are extremely popular and a must-have for any glowy and blinky project.



What sets these LEDs apart is their unique ~~feature~~ feature that allows you to individually address and control the color and brightness of each LED on a strip. Plus, you can (theoretically) chain as many LEDs as you want and control them all using a single microcontroller pin. The more LEDs you chain together, the more fancier your animations will be!

Controlling WS2812B Addressable LEDs with



Did you know?

The popular WS2812 individually addressable RGB LED was first released by [WorldSemi Corporation](#) in late 2013. The company made a name for itself earlier

Estimating Power Requirements

Protecting WS2812B LEDs

Connecting a WS2812B LED Strip to an Arduino

Chaining and Distributing Power
Library Installation

NeoPixel Example Sketches

Example 1 – Simple Color Wipe

Code Explanation:

Example 2 – Rainbow Cycle

Example 3 – Theater Marquee Chasing Lights

Example 4 – Snowflakes

with LED driver solutions, most notably the WS2811, an SOIC chip that turns a common anode RGB LED into a serially controllable LED.

When they stuffed the brains from the WS2811 into a compact package along with a few LEDs, they were able to create what is probably the most common programmable LED lighting solution available today.

Controlling WS2812B Addressable LEDs with

you chain together, the more fancier your animations will be!





with LED driver solutions, most notably the WS2811, an SOIC chip that turns a common anode RGB LED into a serially controllable LED.

When they stuffed the brains from the WS2811 into a compact package along with a few LEDs, they were able to create what is probably the most common programmable LED lighting solution available today.

Controlling WS2812B Addressable LEDs with Arduino

With the WS2812B, you can chain multiple LEDs together. The more LEDs you chain together, the more fancier your animations will be!



with LED driver solutions, most notably the WS2811, an SOIC chip that turns a common anode RGB LED into a serially controllable LED.

When they stuffed the brains from the WS2811 into a compact package along with a few LEDs, they were able to create what is probably the most common programmable LED lighting solution available today.

Cc

ith

with LED driver solutions, most notably the WS2811, an SOIC chip that turns a common anode RGB LED into a serially controllable LED.

When they stuffed the brains from the WS2811 into a compact package along with a few LEDs, they were able to create what is probably the most common programmable LED lighting solution available today.

Cc

ith

with LED driver solutions, most notably the WS2811, an SOIC chip that turns a common anode RGB LED into a serially controllable LED.

When they stuffed the brains from the WS2811 into a compact package along with a few LEDs, they were able to create what is probably the most common programmable LED lighting solution available today.

Cc

ith

with LED driver solutions, most notably the WS2811, an SOIC chip that turns a common-anode RGB LED into a serially-controllable LED.



Cc

ith



with LED driver solutions, most notably the WS2811, an SOIC chip that turns a common-anode RGB LED into a serially-controllable LED.



Cc

ith



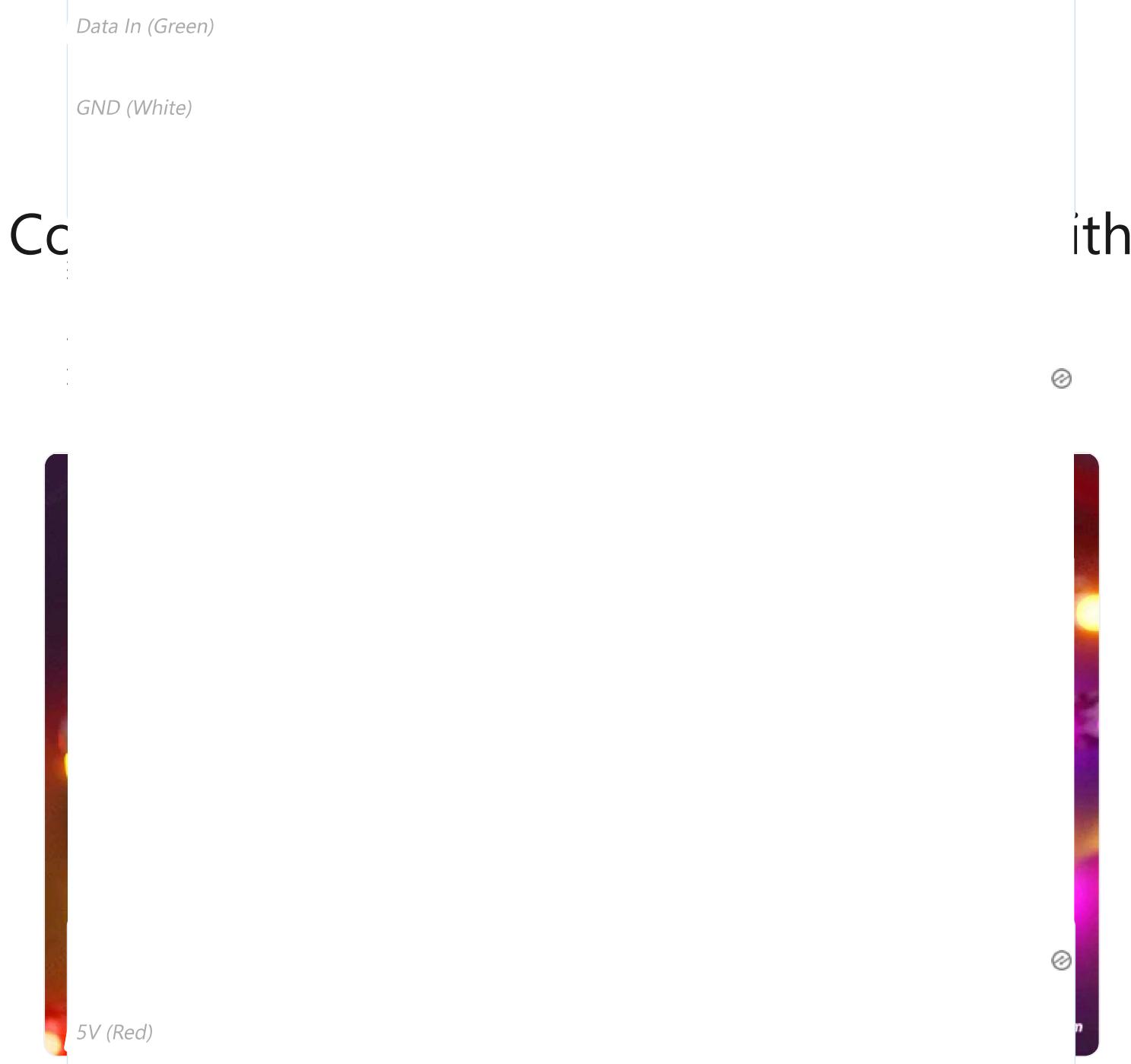
Cc

ith



5V (Red)





Data Out (Green)

Data In (Green)

GND (White)

GND (White)

Cc

ith



5V (Red)



Data Out (Green)

Data In (Green)

GND (White)

GND (White)

Cc

ith



5V (Red)



0

0

Data Out (Green)

Data In (Green)

GND (White)

GND (White)

Cc

ith



5V (Red)



Cc

ith



Cc

ith



Cc

h



Cc

h



Cc

h





Cc



h





Cc



h





Cc

h



Cc

⊗ h



Cc

⊗ h



Cc

⊗ h

Adafruit NeoPixel is not the only library available; there are others, such as the [FastLED library](#), which has more advanced features and supports other LED chipsets.

Cc

⊗ h

Adafruit NeoPixel is not the only library available; there are others, such as the [FastLED library](#), which has more advanced features and supports other LED chipsets.

Cc

h

Adafruit NeoPixel is not the only library available; there are others, such as the [FastLED library](#), which has more advanced features and supports other LED chipsets.

```
// Pin to use to send signals to WS2812B
#define LED_PIN 6

// Number of WS2812B LEDs attached to the Arduino
#define LED_COUNT 12
```

```
#include <Adafruit_NeoPixel.h>

// Pin to use to send signals to WS2812B
#define LED_PIN 6

// Number of WS2812B LEDs attached to the Arduino
#define LED_COUNT 12

// Setting up the NeoPixel library
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
    strip.begin();          // Initialize NeoPixel object
    strip.setBrightness(10); // Set BRIGHTNESS to about 4% (max = 255)
}

void loop() {
    strip.clear(); // Set all pixel colors to 'off'

    // The first NeoPixel in a strand is #0, second is 1, all the way up
    // to the count of pixels minus one.
    for(int i=0; i<LED_COUNT; i++) {
        // Set the i-th LED to pure green:
        strip.setPixelColor(i, 0, 255, 0);

        // strip.show(); sends the new pixel colors to the hardware
    }
}

#define LED_PIN 6

// Number of WS2812B LEDs attached to the Arduino
```

Cc

h

```
void setup() {  
    strip.begin();          // Initialize NeoPixel object  
    strip.setBrightness(10); // Set BRIGHTNESS to about 4% (max = 255)  
}  
  
void loop() {  
    strip.clear(); // Set all pixel colors to 'off'
```

The [WS2812_Definitions.h](#) file defines a huge list of standard colors. In there, you'll find anything from teal to sienna to ghost white. Simply download the file, place it in your sketch folder before opening the sketch, import it into your sketch.

```
#include "WS2812_Definitions.h"
```

And use it to set the color of a pixel, like how the line below changes a pixel's color to teal.

```
strip.setPixelColor(i, TEAL);
```

```
#include <Adafruit_NeoPixel.h>

void setup() {
    strip.begin(); // Initialize NeoPixel object
    strip.setBrightness(10); // Set BRIGHTNESS to about 10% (max = 255)
```

```
// Pin to use to send signals to WS2812B
#define LED_PIN 6
```

```
#define LED_PIN 6  
  
// Number of WS2812B LEDs attached to the Arduino  
#define LED_COUNT 12
```

```
#include <Adafruit_NeoPixel.h>
```

Cc h

```
// Setting up the NeoPixel library  
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);  
// Argument 1 = Number of pixels in the strip  
// Argument 2 = Arduino pin number  
// Argument 3 = Pixel type flags, add together as needed:  
// NEO_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)  
// NEO_KHZ400 400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)  
// NEO_GRB Pixels are wired for GRB bitstream (most NeoPixel products)  
// NEO_RGB Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)  
// NEO_RGBW Pixels are wired for RGBW bitstream (NeoPixel RGBW products)
```

```
strip.begin();
```

```
// Pin to use to send signals to WS2812B  
#define LED_PIN 6
```

```
#define LED_PIN 6

strip.setBrightness(10); // Set BRIGHTNESS to about 4% (max = 255)

#include <Adafruit_NeoPixel.h>

strip.clear();

// Setting up the NeoPixel library
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);
// Argument 1 = Number of pixels in the strip
// Argument 2 = Arduino pin number
// Argument 3 = Pixel type flags, add together as needed:
// NEO_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
// NEO_KHZ400 400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
// NEO_GRB Pixels are wired for GRB bitstream (most NeoPixel products)
// NEO_RGB Pixels are wired for RGB bitstream (v1 FLORA pixels - not v2)
```

Co

h

```
#include <Adafruit_NeoPixel.h>

for(int i=0; i<LED_COUNT; i++) {
    strip.setPixelColor(i, 0, 255, 0);
    strip.show();
    delay(500);
}

// Argument 1 = Number of pixels in the strip
// Argument 2 = Arduino pin number
// Argument 3 = Pixel type flags, add together as needed:
// NEO_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
// NEO_KHZ400 400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)

strip.setPixelColor(i, 0, 255, 0);

strip.setPixelColor(i, 0x00FF00);

strip.setPixelColor(i, 0x00, 0xFF, 0x00);

uint32_t green = strip.Color(0, 255, 0);
strip.setPixelColor(i, green);
```

Co

h

```
strip.setPixelColor(0, 255, 0);

#include <Adafruit_NeoPixel.h>

// Pin to use to send signals to WS2812B
#define LED_PIN 6

// Number of WS2812B LEDs attached to the Arduino
#define LED_COUNT 12

// Setting up the NeoPixel library
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
    strip.begin(); // Initialize NeoPixel object
    strip.setPixelColor(0, 255, 0); // Set pixel #0 to red
}
```

```
strip.setBrightness(10); // Set BRIGHTNESS to about 4% (max = 255)
strip.show();           // Initialize all pixels to 'off'
}

void loop() {
    rainbow(10);
}

// Rainbow cycle along whole strip. Pass delay time (in ms) between frames.
void rainbow(int wait) {
    // 5 cycles of all colors on wheel
    for(long firstPixelHue = 0; firstPixelHue < 5*65536; firstPixelHue += 256) {
        strip.rainbow(firstPixelHue);
```

```
strip.setPixelColor(i, 0, 255, 0);
```

```
#include <Adafruit_NeoPixel.h>
```

```
// Pin to use to send signals to WS2812B
#define LED_PIN 6
```

```
#include <Adafruit_NeoPixel.h>
```

```
// Pin to use to send signals to WS2812B
#define LED_PIN 6
```

```
// Number of WS2812B LEDs attached to the Arduino
#define LED_COUNT 12
```

```
// Setting up the NeoPixel library
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
    strip.begin();          // Initialize NeoPixel object
    strip.setBrightness(10); // Set BRIGHTNESS to about 4% (max = 255)
    strip.show();           // Initialize all pixels to 'off'
}

void loop() {
    // Do a theater marquee effect in various colors...
    theaterChase(strip.Color(255, 255, 255), 50); // White
    theaterChase(strip.Color(255, 0, 0), 50); // Red
    theaterChase(strip.Color(0, 0, 255), 50); // Blue
}

// Theater-marquee-style chasing lights. Pass in a color, and
// a delay time (in ms) between frames
strip.setPixelColor(i, 255, 0);
```

```
void setup() {
    strip.begin();          // Initialize NeoPixel object
    strip.setBrightness(10); // Set BRIGHTNESS to about 4% (max = 255)
    strip.show();           // Initialize all pixels to 'off'
}

void loop() {
    // Do a theater marquee effect in various colors...
    theaterChase(strip.Color(255, 255, 255), 50); // White
    theaterChase(strip.Color(255, 0, 0), 50); // Red
    theaterChase(strip.Color(0, 0, 255), 50); // Blue
}

#include <Adafruit_NeoPixel.h>

// Pin to use to send signals to WS2812B
#define LED_PIN 6

// Number of WS2812B LEDs attached to the Arduino
#define LED_COUNT 12

// Setting up the NeoPixel library
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
    strip.begin();          // Initialize NeoPixel object
    strip.setBrightness(10); // Set BRIGHTNESS to about 4% (max = 255)
    strip.show();           // Initialize all pixels to 'off'
}

void loop() {
    snowflakes(100);
```

```
}

void snowflakes(uint8_t wait) {
    // Setup the pixel array
    int pixel[60];
    for(int p=0; p<60; p++){
        pixel[p] = random(0,255); // Set brightness to about max - 255
    }
    strip.show(); // Initialize all pixels to 'off'
}

void loop() {
    // Do a theater marquee effect in various colors...
    theaterChase(strip.Color(255, 255, 255), 50); // White
    theaterChase(strip.Color(255, 0, 0), 50); // Red
    theaterChase(strip.Color(0, 0, 255), 50); // Blue
}
```

```
void loop() {
    snowflakes(100);
```

```
}

void snowflakes(uint8_t wait) {
    // Setup the pixel array
    int pixel[60];
    for(int p=0; p<60; p++){
        pixel[p] = random(0,255); // Set brightness to about max - 255
    }
    strip.show(); // Initialize all pixels to 'off'
}

void loop() {
    // Do a theater marquee effect in various colors...
    theaterChase(strip.Color(255, 255, 255), 50); // White
    theaterChase(strip.Color(255, 0, 0), 50); // Red
    theaterChase(strip.Color(0, 0, 255), 50); // Blue
}
```

```
void loop() {
    snowflakes(100);
```

```
}

void snowflakes(uint8_t wait) {
    // Setup the pixel array
    int pixel[60];
    for(int p=0; p<60; p++){
        pixel[p] = random(0,255); // Set brightness to about max - 255
    }
    strip.show(); // Initialize all pixels to 'off'
}

void loop() {
    // Do a theater marquee effect in various colors...
    theaterChase(strip.Color(255, 255, 255), 50); // White
    theaterChase(strip.Color(255, 0, 0), 50); // Red
    theaterChase(strip.Color(0, 0, 255), 50); // Blue
}
```

```
void loop() {
    snowflakes(100);
```

```
}

void snowflakes(uint8_t wait) {
    // Setup the pixel array
    int pixel[60];
    for(int p=0; p<60; p++){
        pixel[p] = random(0,255); // Set brightness to about max - 255
    }
    strip.show(); // Initialize all pixels to 'off'
}

void loop() {
    // Do a theater marquee effect in various colors...
    theaterChase(strip.Color(255, 255, 255), 50); // White
    theaterChase(strip.Color(255, 0, 0), 50); // Red
    theaterChase(strip.Color(0, 0, 255), 50); // Blue
}
```

```
void loop() {
    snowflakes(100);
```