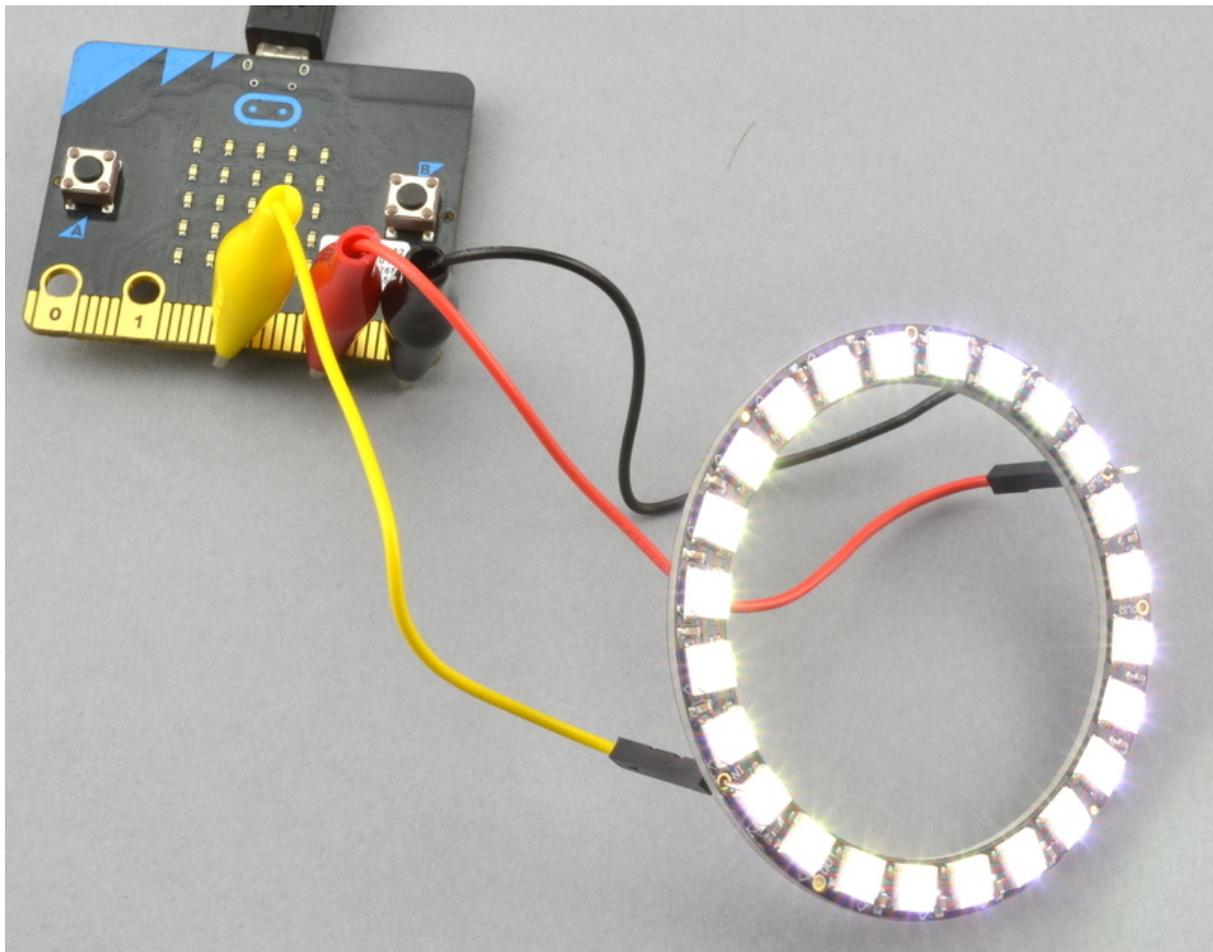




# micro:bit Lesson 3. NeoPixels with micro:bit

Created by Simon Monk



<https://learn.adafruit.com/micro-bit-lesson-3-neopixels-with-micro-bit>

Last updated on 2024-03-08 02:56:36 PM EST

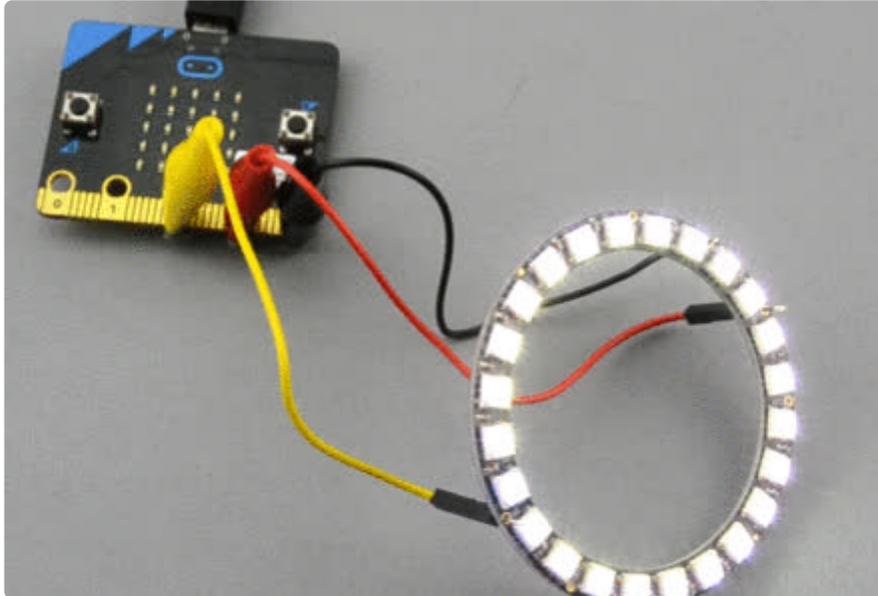
# Table of Contents

|   |   |
|---|---|
| <a href="#">Overview</a>  | 3 |
| <a href="#">Parts</a>   | 3 |
| <a href="#">Connections</a>   | 4 |
| <a href="#">Software</a>  | 5 |
| <ul style="list-style-type: none"><li>• <a href="#">JavaScript Block Code</a></li><li>• <a href="#">Using NeoPixels in the Blocks Editor</a></li><li>• <a href="#">Micro Python</a></li><li>• <a href="#">Arduino</a></li></ul> |   |
| <a href="#">Other Things to Do</a>  | 9 |
| <ul style="list-style-type: none"><li>• <a href="#">About the Author</a></li></ul>  |   |

---

# Overview

NeoPixels (addressable LEDs) are a great way to easily create colorful displays using strips of LEDs or LEDs arranged in a circle like the NeoPixel ring used in this lesson.



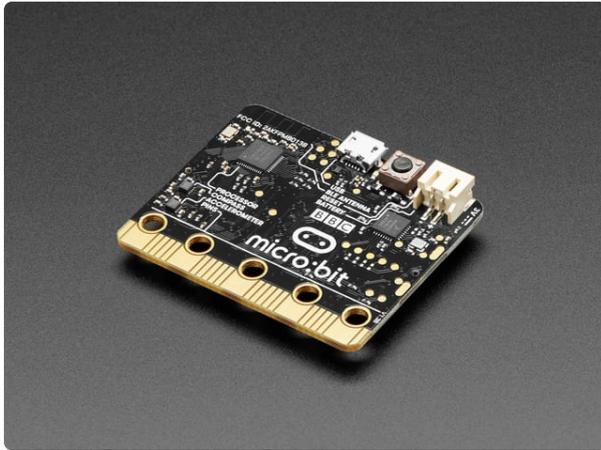
In this lesson you will learn how to connect your NeoPixels to your micro:bit and then how to write software to control it using JavaScript Blocks, MicroPython and Arduino code.

Although this lesson uses a NeoPixel ring, the example programs will work fine with pretty much any NeoPixel display.

---

## Parts

This lesson uses the following parts, but you may choose to use a different NeoPixel display. The micro:bit cannot come with much more than 24 NeoPixels or you may draw too much current from the micro:bit.



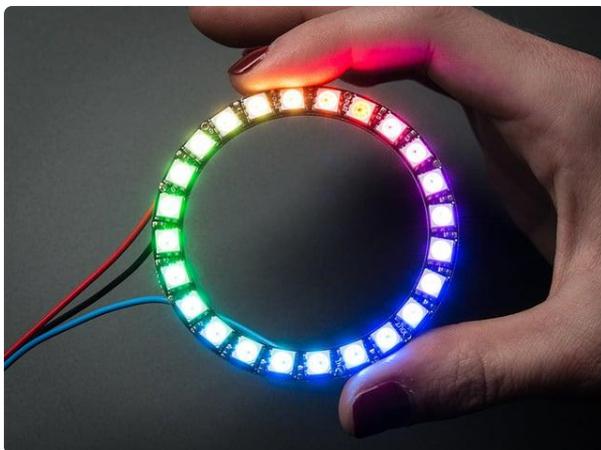
### BBC micro:bit

The British Invasion is here! No, not music...microcontrollers! New to the USA is the newest and easiest way to learn programming and electronics - the BBC...  
<https://www.adafruit.com/product/3530>



### Small Alligator Clip to Male Jumper Wire Bundle - 12 Pieces

For bread-boarding with unusual non-header-friendly surfaces, these cables will be your best friends! No longer will you have long strands of alligator clips that are grabbing little...  
<https://www.adafruit.com/product/3255>



### NeoPixel Ring - 24 x 5050 RGB LED with Integrated Drivers

Round and round and round they go! 24 ultra bright smart LED NeoPixels are arranged in a circle with 2.6" (66mm) outer diameter. The rings are 'chainable' - connect the...  
<https://www.adafruit.com/product/1586>

---

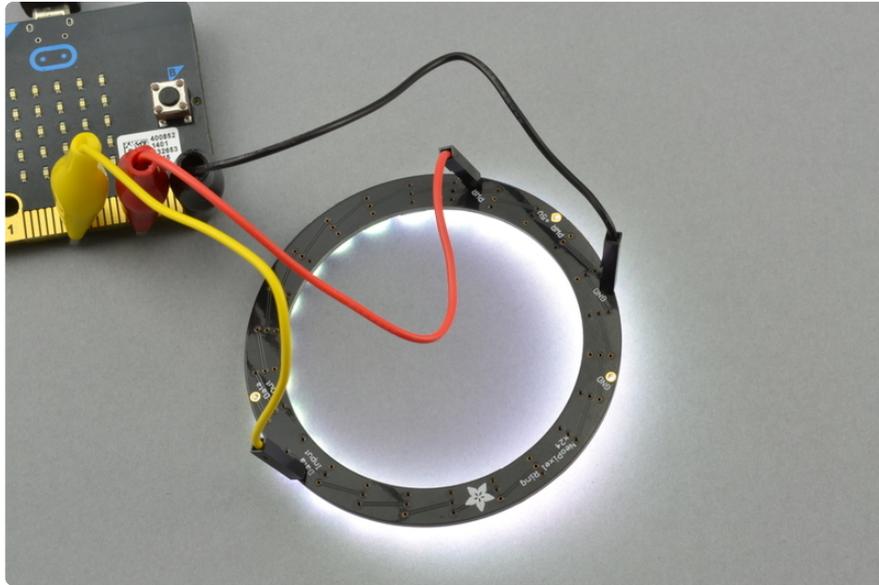
## Connections

NeoPixels are very easy to connect requiring just three connections:

- GND - ground
- Positive supply 3V (in the case of the micro:bit)
- Data Input - data is streamed into the chain of NeoPixels, each NeoPixel passing on the data to the next one along the chain. When the data stops all the

NeoPixels show the color that their data indicates. Connect the Data Input connection on the NeoPixel display to Pin 2 of the micro:bit.

In terms of electrically connecting to the display, if you are using the suggested 24 pixel ring then you can use alligator to male header pin leads to connect up the display by pushing the male header pins into the pins marked GND, PWR and Data In.

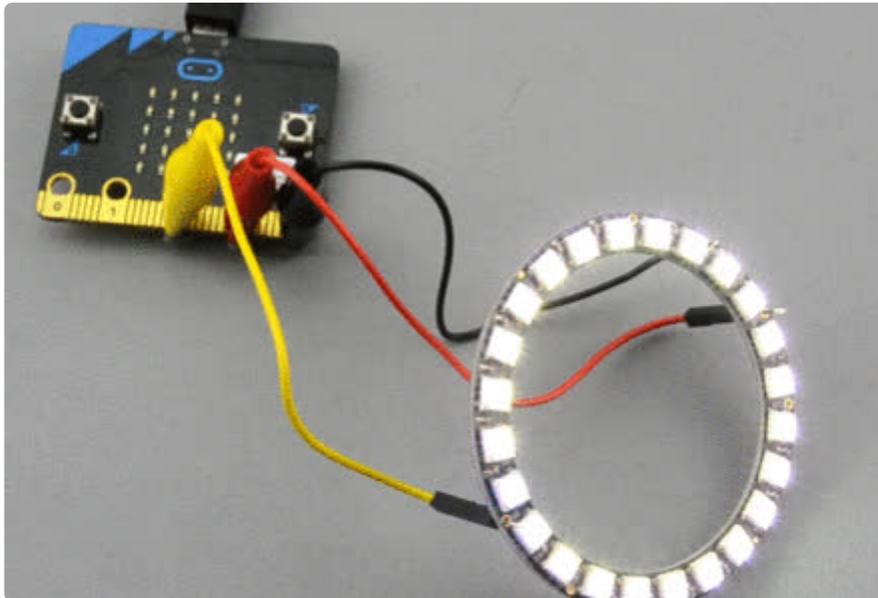


Poking the pins through the holes like this is fine for trying things out, but to make things more permanent, solder the pins from the top side of the board and clip off the excess pin.

---

## Software

NeoPixel displays allow you to set the color and brightness of every pixel in the display independently. The following example program uses just two colors: one as a background the other color being the color for just one pixel that will rotate round from one pixel to the next.



Don't overload your micro:bit. With a 24 pixel display, like this, your micro:bit will only be able to supply enough current to light all the pixels at half maximum brightness, so, keep the numbers low to avoid problems. Some older micro:bits had a voltage regulator that did not automatically protect the micro:bit against overcurrent, so if you have one of these and light too many pixels too brightly, you could destroy your micro:bit.

## JavaScript Block Code

The JavaScript Blocks Code editor is embedded directly on this page below. From the editor, you can click on Download button (bottom right) and then copy the downloaded file onto your micro:bit. Alternatively, you can [Click here \(https://adafru.it/Cjo\)](https://adafru.it/Cjo) to open the editor in a separate browser tab.

The Blocks code works like this:

First, in the on start block a variable ring is assigned to the NeoPixel display. This specified the pin that will be used to send data to the NeoPixel and also the number of pixels in the display.

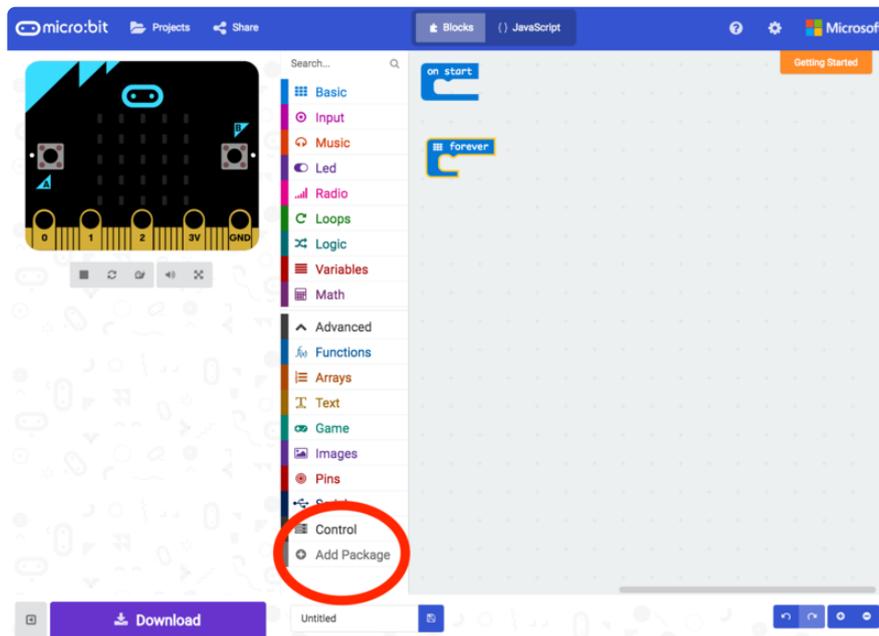
The forever block contains a for loop that counts from 0 to 23, keeping the counter in the variable index.

Inside the for loop, the pixel at the index position is set to blue. The show block is needed to actually send the updated pixel color to the pixel ring. After a pause to 50 milliseconds, the pixel at position index is then set to be a color defined by its red,

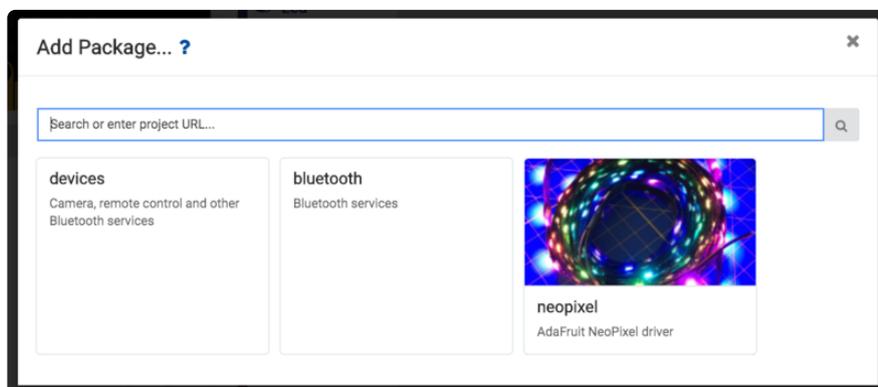
green and blue components. As all three values are equal, the background color will be white, but not a very bright white as the value is only 16 (out of 255).

## Using NeoPixels in the Blocks Editor

If you want to make your own project that uses NeoPixels, before you can use NeoPixels with the Blocks Editor, you will need to add Adafruit's NeoPixel package to your environment. To do this, click on the Add Package option circled below.



This will open the dialog below from which you should select the neopixel package. Once you have done this you will find an extra category of blocks called Neopixel appears below Math.



# Micro Python

To run the MicroPython version of the code, open up the [online Python editor here \(https://adafru.it/CiK\)](https://adafru.it/CiK) and paste the following code into the editor window.

```
from microbit import *
from neopixel import NeoPixel

num_pixels = 24
foreground = [0x00, 0x00, 0xFF] # Hex color - red, green and blue
background = [0x10, 0x10, 0x10]

ring = NeoPixel(pin2, num_pixels)

while True:
    # blue dot circles around a white background (for PixelRing 24)
    for i in range(0, num_pixels):
        ring[i] = foreground # set pixel i to foreground
        ring.show()         # actually display it
        sleep(50)           # milliseconds
        ring[i] = background # set pixel to background before moving on
        ring.show()
```

The code defines three variables: num\_pixels (the number of pixels in the display, foreground and background, the two colors are specified as standard red, green and blue hexadecimal numbers. If you prefer to use decimal numbers to specify the color, then you could change these lines to be:

```
foreground = [0, 0, 255] # Hex color - red, green and blue
background = [16, 16, 16]
```

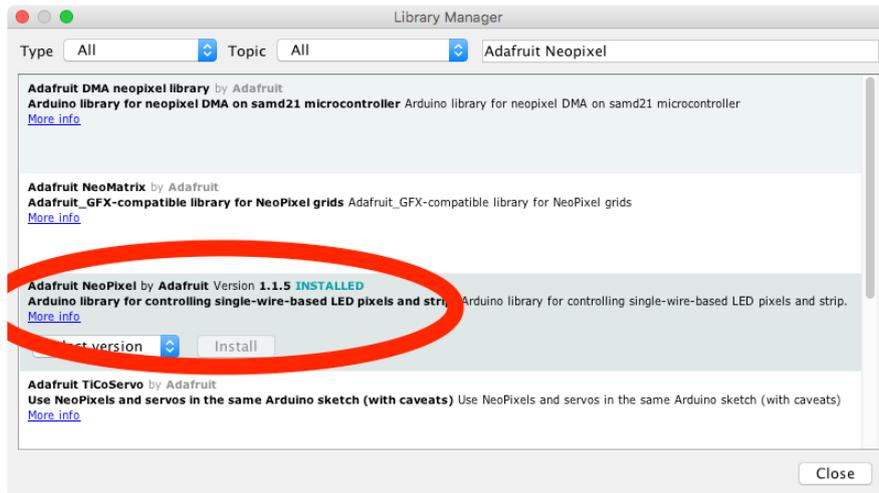
For a discussion on different ways of specifying colors, see [https://en.wikipedia.org/wiki/Web\\_colors](https://en.wikipedia.org/wiki/Web_colors) (<https://adafru.it/Cjp>)

# Arduino

Make sure that you have your Arduino environment set up for micro:bit by following [this guide \(https://adafru.it/CiL\)](https://adafru.it/CiL).

Before you can use NeoPixels with a micro:bit programmed from the Arduino IDE, you will need to make sure that your Arduino IDE has a recent version of the Adafruit NeoPixel library installed. To do this select Sketch menu -> Include Library -> Manage Libraries..

From the Library Manager, search for Adafruit Neopixel and then select and install the library called just Adafruit Neopixel by Adafruit.



Now start a new Sketch by clicking on the File menu and New. Then paste the following code into the editor window.

```
#include <Adafruit_NeoPixel.h>

const int numPixels = 24;
const int pixelPin = 2;

Adafruit_NeoPixel ring = Adafruit_NeoPixel(numPixels, pixelPin);
uint32_t foreground = ring.Color(0x00, 0x00, 0xFF); // r, g, b - blue
uint32_t background = ring.Color(0x10, 0x10, 0x10); // dim white

void setup() {
  ring.begin(); // start the NeoPixel display
}

void loop() {
  // blue dot circles around a white background (for PixelRing 24)
  for (int i = 0; i < numPixels; i++) {
    ring.setPixelColor(i, foreground); // set pixel i to foreground
    ring.show(); // actually display it
    delay(50); // milliseconds
    ring.setPixelColor(i, background); // set pixel to background before moving on
    ring.show();
  }
}
```

For a description of how the code works, take a look at the description in the MicroPython section. The programs are very similar.

---

## Other Things to Do

Try changing the foreground and background colors of the program.

You can also try reducing the delay/pause period to speed up the rotating effect.

There are loads of interesting effects you can make with NeoPixels, so take a look at Lady Ada's [NeoPixel Über Guide \(https://adafru.it/dhw\)](https://adafru.it/dhw)

# About the Author

If you have found this lesson useful, and want to learn Python, then you might like my book Programming the BBC micro:bit: Getting Started with MicroPython.

