

Adafruit NeoPixel Library

build passing

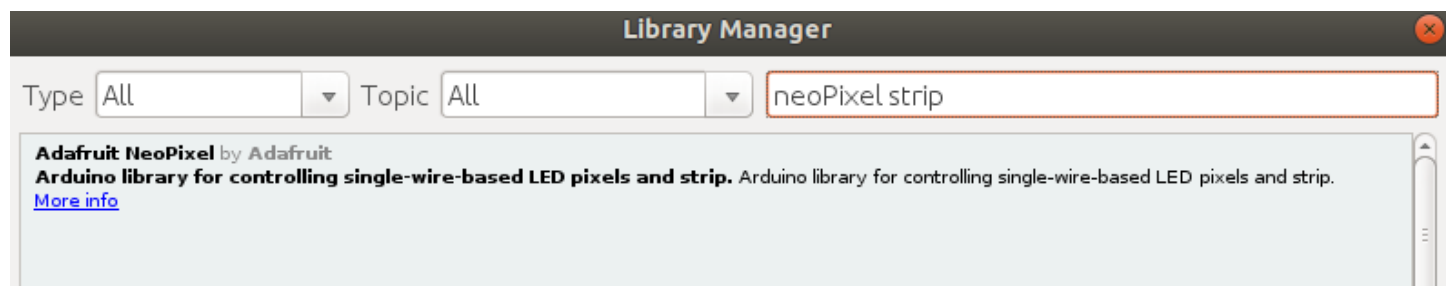
Arduino library for controlling single-wire-based LED pixels and strip such as the [Adafruit 60 LED/meter Digital LED strip](#), the [Adafruit FLORA RGB Smart Pixel](#), the [Adafruit Breadboard-friendly RGB Smart Pixel](#), the [Adafruit NeoPixel Stick](#), and the [Adafruit NeoPixel Shield](#).

After downloading, rename folder to 'Adafruit_NeoPixel' and install in Arduino Libraries folder. Restart Arduino IDE, then open File->Sketchbook->Library->Adafruit_NeoPixel->strandtest sketch.

Compatibility notes: Port A is not supported on any AVR processors at this time

Installation

First Method



1. In the Arduino IDE, navigate to Sketch > Include Library > Manage Libraries
2. Then the Library Manager will open and you will find a list of libraries that are already installed or ready for installation.
3. Then search for Neopixel strip using the search bar.
4. Click on the text area and then select the specific version and install it.

Second Method

1. Navigate to the [Releases page](#).
2. Download the latest release.
3. Extract the zip file
4. In the Arduino IDE, navigate to Sketch > Include Library > Add .ZIP Library

Features

- Simple to use

Controlling NeoPixels "from scratch" is quite a challenge, so we provide a library letting you focus on the fun and interesting bits.

- **Give back**

The library is free; you don't have to pay for anything. Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

- **Supported Chipsets**

We have included code for the following chips - sometimes these break for exciting reasons that we can't control in which case please open an issue!

- AVR ATmega and ATtiny (any 8-bit) - 8 MHz, 12 MHz and 16 MHz
- Teensy 3.x and LC
- Arduino Due
- Arduino 101
- ATSAM21 (Arduino Zero/M0 and other SAMD21 boards) @ 48 MHz
- ATSAM51 @ 120 MHz
- Adafruit STM32 Feather @ 120 MHz
- ESP8266 any speed
- ESP32 any speed
- Nordic nRF52 (Adafruit Feather nRF52), nRF51 (micro:bit)

Check forks for other architectures not listed here!

- **GNU Lesser General Public License**

Adafruit_NeoPixel is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Functions

- `begin()`
- `updateLength()`
- `updateType()`
- `show()`
- `delay_ns()`
- `setPin()`
- `setPixelColor()`

- fill()
- ColorHSV()
- getPixelColor()
- setBrightness()
- getBrightness()
- clear()
- gamma32()

Examples

There are many examples implemented in this library. One of the examples is below. You can find other examples [here](#)

Simple

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
    #include <avr/power.h>
#endif
#define PIN        6
#define NUMPIXELS 16

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
#define DELAYVAL 500

void setup() {
    #if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
        clock_prescale_set(clock_div_1);
    #endif

    pixels.begin();
}

void loop() {
    pixels.clear();

    for(int i=0; i<NUMPIXELS; i++) {

        pixels.setPixelColor(i, pixels.Color(0, 150, 0));
        pixels.show();
        delay(DELAYVAL);
    }
}
```

Contributing

If you want to contribute to this project:

- Report bugs and errors
- Ask for enhancements
- Create issues and pull requests
- Tell others about this library
- Contribute new protocols

Please read [CONTRIBUTING.md](#) for details on our code of conduct, and the process for submitting pull requests to us.

Roadmap

The PRIME DIRECTIVE is to maintain backward compatibility with existing Arduino sketches -- many are hosted elsewhere and don't track changes here, some are in print and can never be changed!

Please don't reformat code for the sake of reformatting code. The resulting large "visual diff" makes it impossible to untangle actual bug fixes from merely rearranged lines. (Exception for first item in wishlist below.)

Things I'd Like To Do But There's No Official Timeline So Please Don't Count On Any Of This Ever Being Canonical:

- For the show() function (with all the delicate pixel timing stuff), break out each architecture into separate source files rather than the current unmaintainable tangle of #ifdef statements!
- Please don't use updateLength() or updateType() in new code. They should not have been implemented this way (use the C++ 'new' operator with the regular constructor instead) and are only sticking around because of the Prime Directive. setPin() is OK for now though, it's a trick we can use to 'recycle' pixel memory across multiple strips.
- In the M0 and M4 code, use the hardware systick counter for bit timing rather than hand-tweaked NOPs (a temporary kludge at the time because I wasn't reading systick correctly). (As of 1.4.2, systick is used on M4 devices and it appears to be overclock-compatible. Not for M0 yet, which is why this item is still here.)
- As currently written, brightness scaling is still a "destructive" operation -- pixel values are altered in RAM and the original value as set can't be accurately read back, only approximated, which has been confusing and frustrating to users. It was done this way at the time because NeoPixel timing is strict, AVR microcontrollers (all we had at the time) are limited, and assembly language is hard. All the 32-bit architectures should have no problem handling nondestructive brightness scaling -- calculating each byte immediately before it's sent out the wire, maintaining the original set value in RAM -- the work just hasn't been done. There's a fair chance even the AVR code could manage it with some intense focus. (The DotStar library achieves nondestructive brightness scaling because it doesn't have to manage data timing so carefully...every architecture, even ATtiny, just takes whatever cycles it needs for the multiply/shift operations.)

Credits

This library is written by Phil "Paint Your Dragon" Burgess for Adafruit Industries, with contributions by PJRC, Michael Miller and other members of the open source community.

License

Adafruit_NeoPixel is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. Adafruit_NeoPixel is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the [GNU Lesser General Public License](#) for more details. You should have received a copy of the GNU Lesser General Public License along with NeoPixel. If not, see [this](#)