

A Guide to Addressable LEDs

Addressable LEDs (Disambiguation)

- Neopixels – Brand Name
- WS281[1/2/3][A/B/C] – Technical Name
 - Different Numbers Refer to Generations, 1 is old, 3 is better but 2 is most common.
- WS2812 – Most Common, what you'll hear me call them.

WS2812 - Packages

- WS2812 actually refers to the chip that controls the LED.
- The chip is embedded in various LED packages.
 - SMD 5050 – 5mm square surface mount.
 - SMD 3535 – 3.5mm square surface mount.
 - Through Hole – Normal LED Package.

Buying WS2812

- Amazon – Quick Delivery, Good Choice, More Expensive
- AliExpress/eBay – Slow Delivery, Better Choice, Cheaper
- [A good example on amazon.](#)
- It is best to search for ‘WS2812’ to make sure the tape you find is addressable.

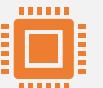
LED Tape

- Most Common Format.
- Easy to work with.
- Don't trust Supplied Adhesive.
- IP Ratings
 - IP30 – No Protection.
 - IP65 – Plastic/Rubber Coating on top of strip.
 - IP67 – IP30 strip inside a sealed rubber tube.
 - More expensive but gives you the option to remove waterproofing if necessary.
 - Easier to solder than IP65.
- Watch out for strips with separate Red, Green and Blue LEDs.
- Watch out for strips that are 12v. These are fine to use but need a separate power supply to Arduino.

Other Form Factors

- WS2812s also come in other form factors such as rings and grids which can be useful for specific projects.
- Other form factors will connect just like strips and can actually be used and connected together.

Connecting to Addressable LEDs



Each LED in the chain has 4 pins.

5v
GND
Data in
Data Out



To connect a set of LEDs we need the 5v, Ground and Data in Pins.



One end of LED tape will only have 5v, GND and Data in.



The other end will have 5v, GND and Data Out for connecting to another set of LEDs.



The Data In pin and ground must be connected to the Arduino for data to get through.

Powering LEDs

- Addressable LEDs predominantly run on 5v so when using a small number it is okay to power them from the 5v plug on your Arduino.
- LEDs use different amounts of power depending on how much brightly they are lit.
- When the Red, Green and Blue LEDs are all lit at full brightness, each pixel uses about 60mA. In normal use however it is rare for this to happen.
- I usually use the 60mA figure to estimate the total maximum power draw of the strip.

Powering LEDs II

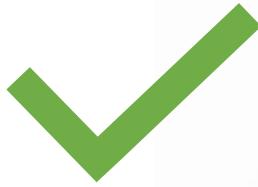
- If we have a meter of LED Tape at 60 LEDs/m so 60 LEDs, we can estimate the total current as $(60\text{LEDs} * 60\text{mA}) / 1000$ (to get a result in Amps) = 3.6A
- $3.6\text{A} * 5\text{v} = 18\text{W}$ – When selecting a power supply, they will usually note the voltage and wattage so it is useful to know the Watts (power) when looking for one.
- Based on this then, and given the usual USB max current of about 1A, we can estimate that when powering the LEDs from the Arduino we don't want to connect more than 16 LEDs. If you know for sure however that you won't be lighting them all at once then you could get away with closer to 30.

Powering LEDs III

- The only problem with the LEDs running at 5v, is that over a relatively short distance the LEDs will start to dim due to voltage drop.
- This problem becomes much more apparent after more than 100 LEDs so below that it isn't much of an issue.
- The simple solution is to connect the same power supply at different points in the LED Strip so as to even out the voltage across the whole length.

Powering LEDs III

- When buying a power supply you will usually have a choice between a power supply that you must wire one yourself, and one that comes as a contained unit.
- To begin with, I recommend self contained units as they are much safer than the other units but after a certain amount of LEDs, an open unit is the only option.
- Once the number of LEDs is in the thousands, other solutions start to make sense such as converting AC to 24DC and then down to 5V at various points along the strip. This is usually only necessary in the most demanding situations such as full walls coated in LED tape.



Power Supply Designs

Controlling LEDs

- Like with everything else, there are different ways to control these LEDs. You can either control them with the Arduino itself, or from an external program over the network such as MAX, OpenFrameworks or Resolume.
- The two options I'm going to note today are :
 - FastLED Library for Arduino
 - ESPixelstick firmware for ESP8266

Arduino FastLED Library

- Installs just Like any other Library, from the Library Manager.
- Comes with Examples!
- Control LEDs by changing variables in an array with one element per pixel.
- I recommend making a function for each animation and swapping them out based on Serial Messages.

ESPPixelstick Firmware for ESP8266

- Installs differently from normal.
- Installer is a separate file that you run to install the firmware.
- Input the Wifi name and Password and Upload.
- The ESP will then reboot and connect to the Wifi, it will send its IP on the serial port.
- It receives data for each pixel via sACN protocol which can be generated in a similar manner to FastLed in openFrameworks.
- sACN can also be generated from video in programs like Resolume and Madrix.
- <https://github.com/forkineye/ESPPixelStick>

Addressable LEDs Vs Multiplexing

- Addressable LEDs can be on simultaneously.
- Addressable LEDs require less code.
- A huge number of Addressable LEDs can be controlled from one pin.
- Multiplexing draws less power as only one LED is ever lit at a time.
- Multiplexing is most commonly used to drive seven segment displays.
- Multiplexing RGB LEDs adds an extra layer of complexity.

