VHS WLED & Pixels Workshop

Introduction to WLED and RGB LED Pixels..

A workshop brought to you by the fine folks at VHS (Vancouver Hack Space)

Jan 2023 packetbob@gmail.com

WLED

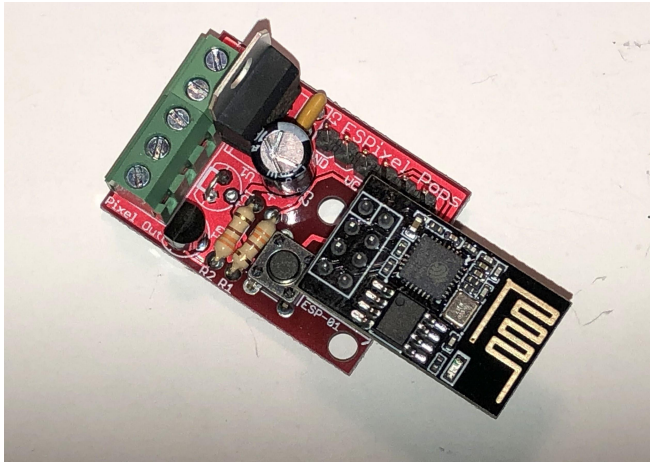
WLED is open source software that controls addressable LEDs. It lets you use a web page to program static and animation effects with your LEDs. It also supports various ways of remotely controlling the LEDs via E1.31 (DMX over Ethernet), MQTT & Philips HUE to name a few. Some features include:

- Over 100 various effects that often have multiple parameters to adjust
- Configurable via web page or stand alone IOS/Android app
- Can connect to existing Wi-Fi network or act as stand-alone AP for connectivity
- Can be set to run stand alone effects for props and costumes
- Runs on most ESP8266 and ESP32 variants (Sound reactive version available for ESP32)
- Supports multiple addressable LED types and allows for over 500 attached LEDs
- Can segment LEDs into sections and run separate effects in each
- Pre-compiled binaries available for various platforms so you don't need to do any programming

Github Page: https://github.com/Aircoookie/WLED
Support Wiki: https://kno.wled.ge/
Discourse Site: https://wled.discourse.group/
GitBook Site: https://bist.gitbook.io/wled-project/

ESPixelPOPs WLED Kit

Included in this workshop is a kit to build a Wi-Fi enabled, ESP8266 based, WLED pixel controller and a strip of WS2812 pixels to enable you to get started with addressable RGB LED control. The WLED firmware has already been loaded on the ESP-01 controller.

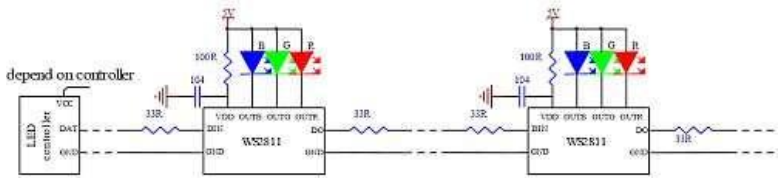


The ESPixelPOP WLED kit is designed to be compact and uses all through hole components for ease of soldering

The WLED firmware can run on any ESP8266 or ESP32 variant. The ESPixelPOP uses a ESP-01 controller which allows for a very small form factor. Due to the smaller Flash memory size in the ESP-01 it does not support OTA firmware updates but does support all the other features (and can be updated via the programming header). The ESPixelPOP PCB also provides a voltage regulator for the 3.3V the ESP-01 requires and a 2N7000 MOSFET based level translator to provide a 5V level data line to the pixels from the 3.3V data out from the ESP8266.

Pixel Basics

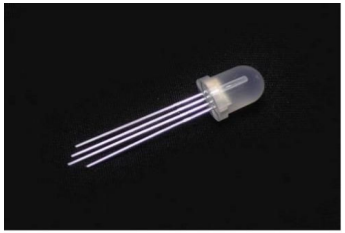
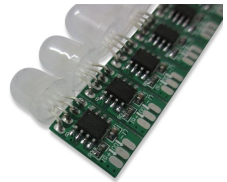
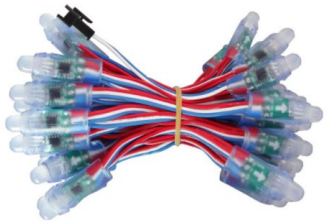
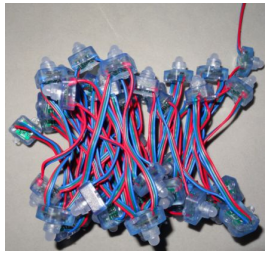
- A Pixel consists of 3 LEDS (RGB) and a controller IC.
- The Pixel controller IC reads the serial input stream, adjusts its RGB LED PWM output, strips its RGB data from the stream and then send the remaining stream to the next pixel. Each pixel does this in line till the end where there is a single RGB value left for the last pixel in the string.

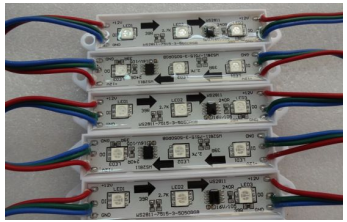


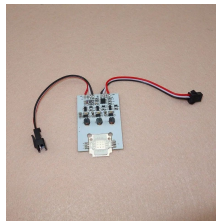
- Adafruit's NeoPixel and the FastLED project are free, well supported and very popular libraries for both Arduino and ESP8266 variant microcontrollers.

Types Of Pixels





Notes

- A PDF with a build guide and basic configuration info can be found here:
- https://zappedmyself.com/vhs/vhs-wled-pixel-controller-kit/

- When WLED first boots up it will default to AP mode, the SSID name and pw is WLED-AP-xx (matches what your kit is labeled). Once connected point your browser to: 4.3.2.1.

- You can leave the WLED controller in AP mode (and just connect using the SSID & PW) or connect it to your Wi-Fi network. Note that it has no PW protection when connected in a Wi-Fi network.

- You can use any web browser (or download an IOS/Android app for your phone) to control and configure WLED.

- If WLED can't connect to the configured Wi-Fi network it will revert to AP mode. The options for this can be configured.

- WLED does have a current limiting function but this is just a calculation that limits the max intensity of LEDs to keep the estimated current under the limit you set. It's not foolproof. YOu can still overload your power supply if there is a software glitch that max's out brightness. Best to have a power supply that can handle all the LEDS at max brightness (approx 60 mA per pixel)

More Notes

- When WLED boots up it`s default settings will light up its connected pixels as orange. It will always do this unless you configure a preset and change the settings to start with this preset. If your LEDS are not orange, then you can change the color order, pixel type and pixel count.
- The presets can be stored in a playlist that WLED can also be set to start playing.
- You can set up the string of LEDS to have multiple segments and each segment can have different effects.
- You can backup the presets and configuration files.
- The limitation of the ESP-01 version of the processor used in this kit is that the software doesn`t support OTA updates. To update the firmware you need to use a USB-Serial adapter. Information is in the build guide. I have loaded on the most recent release (13.3).
- While the ESP8266 used in our controller doesn’t support sound reactive control, you can use a PC app called LedFX. It can control multiple WLED based devices based on an audio stream from the PC.

Pixel Display Tips

- Very handy to have either a stand alone pixel tester (can be purchased or you can configure an uC to send pixel data sequences) or a spare controller that you use to send data from. Also handy to have know good test Pixels that you can connect to your controllers to ensure they are functional
- Good idea to break large displays into smaller parts and do a burn-in test prior to final assembly.
- Pixels don't seem to have great quality control and some will fail. It's a good idea to design and build your displays with replacing pixels in mind.
- Waterproofing can be really be an issue in our NW winters. Strips do not last long outside. Bullets are better. Any will tend to fail if stress is put on wires. Best to try to protect from elements as much as possible.
- The WS281x chips like a 5V data signal so best to use a driver/buffer if your uC is only 3.3V (ESP variants).
- Ensure you have enough power for the number of LEDs you have connected. At max intensity each pixel can use 60 mA and this can add up quickly. The 30 LEDs in the supplied kit can draw 1.8 A at full brightness.