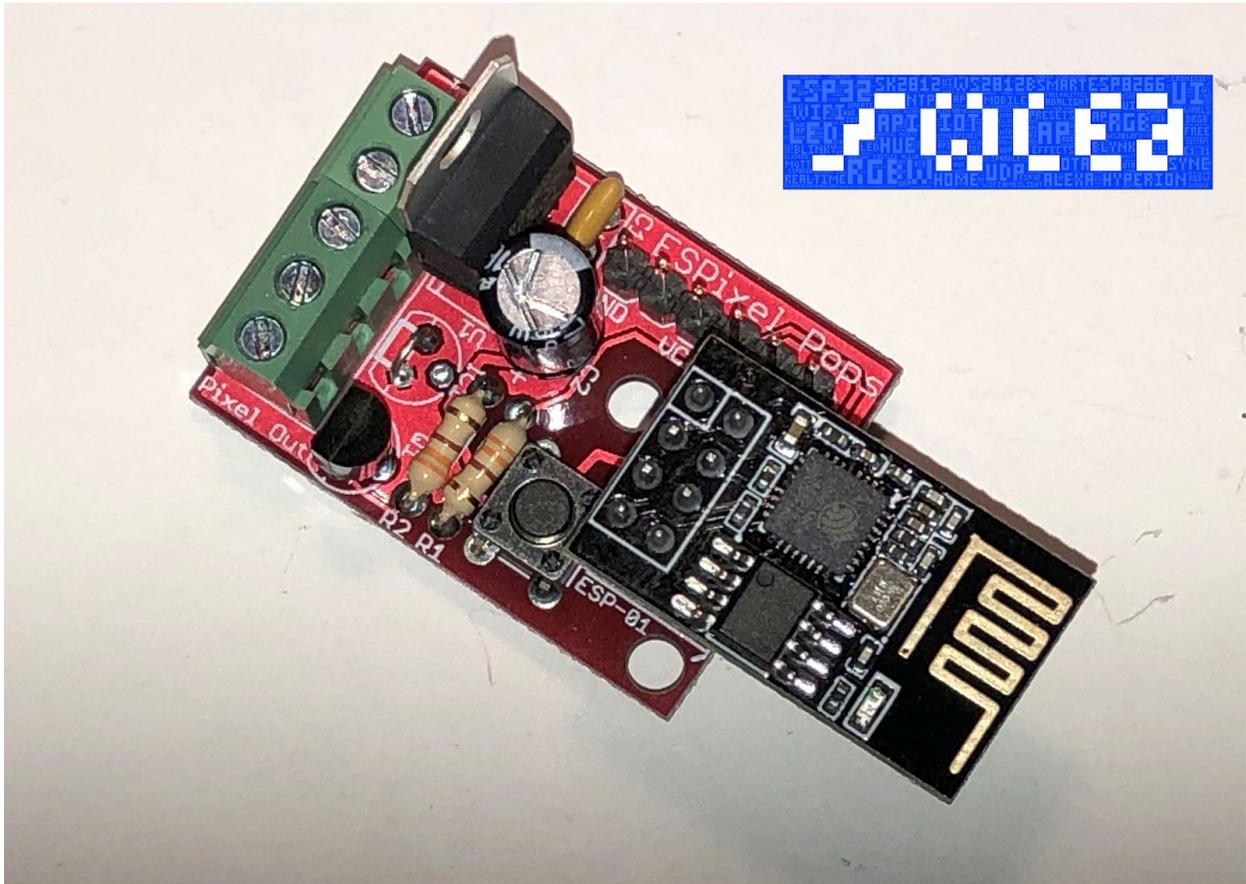


ESPixelPOPs WLED Guide v5.16.2020



Your [ESPixelPOPs](#) board runs the [ESPixelStick firmware](#) and this is fantastic software for a E1.31 Pixel controller (it's designed purpose). But let's say you need some lighting effects and don't want to have the complexity of running a PC with xLights or Vixen (or a Falcon Player).

Now you can certainly use the Effects feature of ESPixelStick firmware but what if you want something with more features...

You can find loads of Arduino/ESP8266 based lighting controllers on the Internet but what if you want to use your ESPixelPOPs...

Well you can get the following features using some firmware called [WLED](#). It gives you a web based control panel (also supports a IOS and Android app) to run all sorts of lighting effects in a stand alone mode (so you don't need to stream E1.31 data to it). You are not going to run your sequenced Xmas lighting display from it but you may find it handy for small lighting projects. Some features include:

- WS2812FX library integrated for over 100 special effects
- FastLED noise effects and 50 palettes
- Modern UI with color, effect and segment controls
- Segments to set different effects and colors to parts of the LEDs
- Settings page - configuration over network
- Access Point and station mode - automatic failsafe AP
- Support for RGBW strips
- 16 user presets to save and load colors/effects easily, supports cycling through them.
- Macro functions to automatically execute API calls
- Nightlight function (gradually dims down)
- Full OTA software updatability (HTTP + ArduinoOTA), password protectable
- Configurable analog clock + support for the Cronixie kit by Diamex
- Configurable Auto Brightness limit for safer operation

This guide will show you how to install and set up the WLED firmware on your ESPixelPOPs controller. While you can download the full source code for WLED, for this guide we will use the precompiled binary files.

I have put together this guide based on my experiences with multiple ESPixelPOP builds and while I expect it is all correct, there is no guarantee. If you find any errors or something that should be added please let me know and I will revise it.

packetbob@gmail.com

You will need the following tools and equipment:

- Completed and functional ESPixelPOPs board
- USB to Serial TTL Adapter (3.3V or 5V Version)
- 5V Power Supply
- Laptop or PC with Wi-Fi
- WS281x Pixels

If you want more information on the ESPixelPOPs project check this page:

https://www.doityourselfchristmas.com/wiki/index.php?title=ESPixel_Stick_%26_ESPixel_Pops

If you want a build guide for you ESPixelPOPs board check here:

<http://zappedmyself.com/projects/espixelpops/>

Step #1 - Determine what WLED firmware you want.

There is one caveat to running the WLED firmware on your ESPixelPOPs. The ESP-01 module used in the ESPixelPOPs doesn't have enough memory to load the full feature set of the WLED firmware. Luckily the developer has provided various binary files for differing sizes of flash.

If you have a 1 Meg Flash ESP-01 module (typically black) then you have two options:

- If you want all the options except being able to update the firmware over the web connection then install the **WLED_0.x.x_ESP8266_1M_full.bin** version.
- If you want to be able to update the firmware over the wireless connection but don't need the extra interfaces (like Alexa, Blynk, Infrared, etc), then install the **WLED_0.x.x_ESP8266_1M_ota.bin** version.

If you have a 512K Flash ESP-01 module (typically blue) then you need to install the **WLED_0.x.x_ESP8266_512k.bin** version. (Note that this version may soon be unsupported)

You can get further information on the various WLED firmware versions:

<https://github.com/Aircoookie/WLED/wiki/Install-WLED-binary#what-binary-should-i-use>

Step #2 - Download the firmware.

- ❑ Create a folder on your PC called WLED.
- ❑ Go to this page (<https://github.com/Aircoookie/WLED/releases>) and download the relevant firmware file (as noted above) to the WLED folder you previously created.

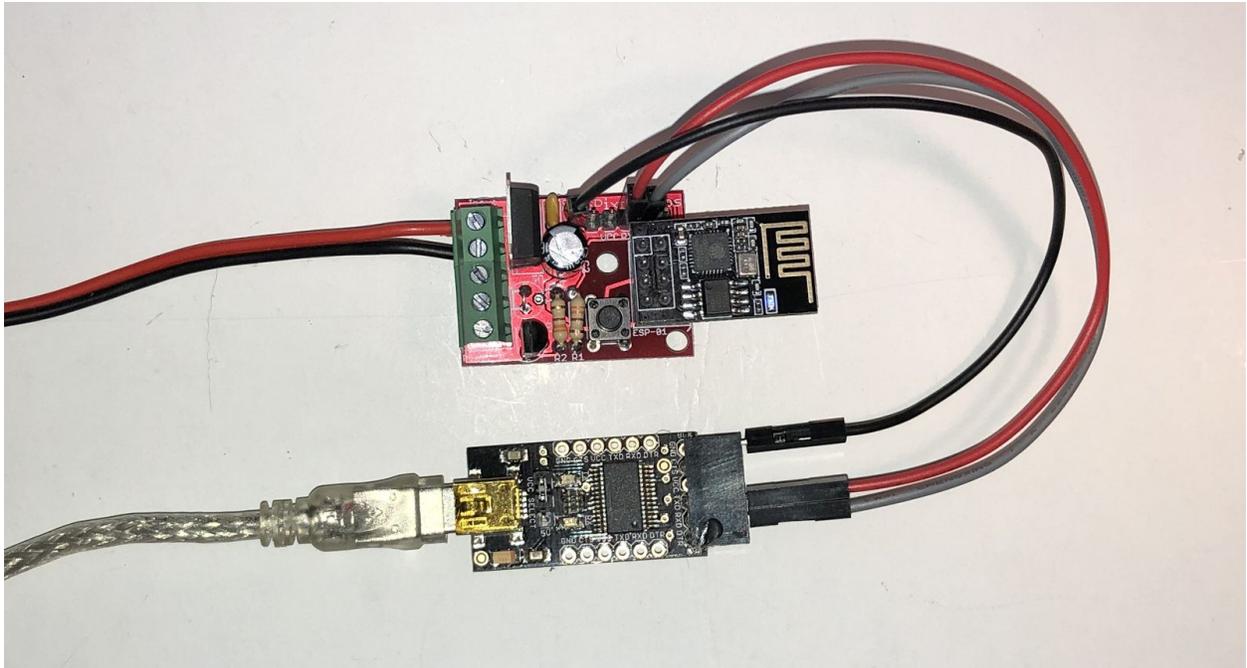
Step #3 - Download the ESP programming utility.

As we are using the precompiled binary (.bin file) we need a way to load it into the ESP-01 module. We will use a flash loader application called [ESPHome-Flasher](#) that has both a Windows and a MacOS version. Note that you don't need to install this application but can just double click it in the folder you have downloaded it to.

- ❑ Go to this page (<https://github.com/esphome/esphome-flasher/releases>) and download the required flash loader. Store it in the same WLED folder where you saved the firmware.

Note that this flash utility is a bit clunkier than the one used for the ESPixelStick firmware

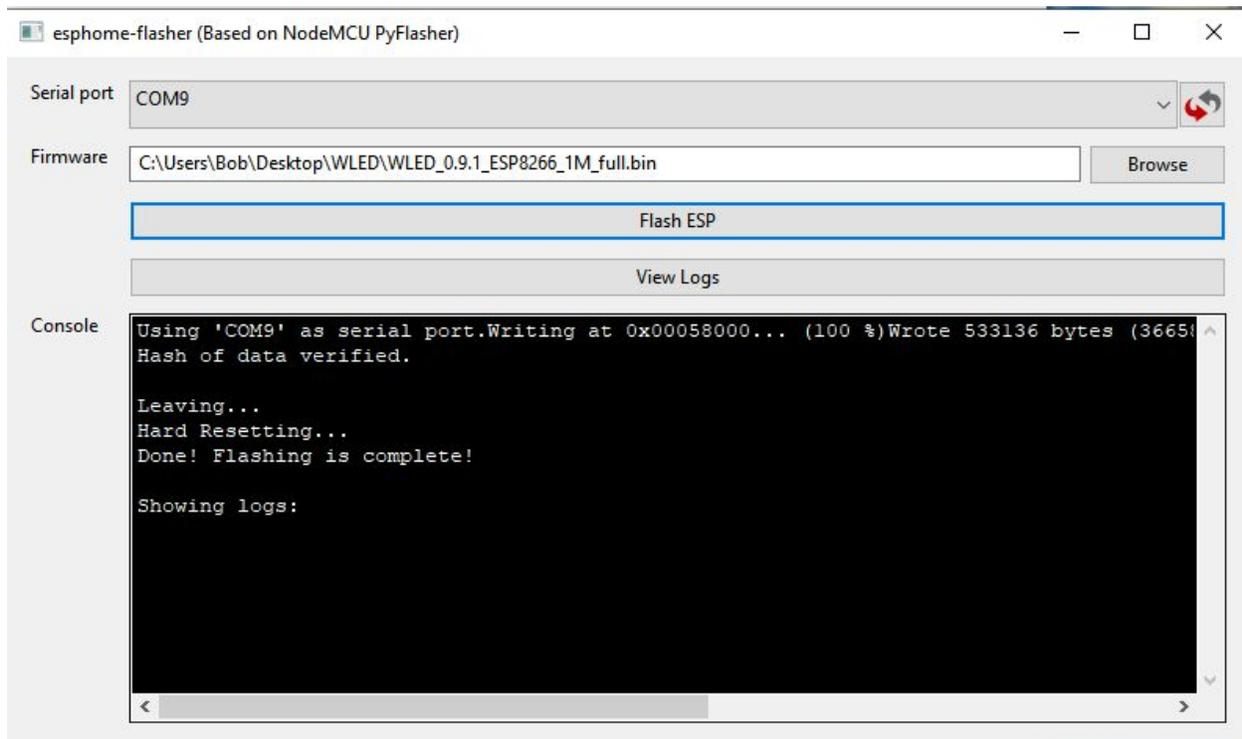
Step #4 - Connect your USB-TTL adapter and confirm connectivity



- Connect your USB-TTL adapter to your PC and ensure it is recognized by your PC.
- Connect your USB-TTL adapter pins to the header pins on the ESPixelPOP as noted:
 - USB Adapter GND to PixelPOP GND
 - USB Adapter TXD to PixelPOP RX
 - USB Adapter RXD to PixelPOP TX
- Open up the WLED folder on your PC you created previously and launch the ESPHome-Flasher application.
- Ensure the Serial Port dropdown box has the correct port selected for your USB-TTL adapter (You can refresh the list of available serial port devices using the button to the right of the drop down).
- Select the View Logs button.
- Power up the ESPixelPOPs board.
- If your ESP-01 module was previously loaded with the ESPixelStick firmware you should see its bootup messages.
- If your ESP-01 module has no firmware loaded you should see a `ready` message.
- If you don't see any messages in the Console window, double check your USB-TTL connections to the ESPixelPOPs board.
- Close the ESPHome-Flasher application

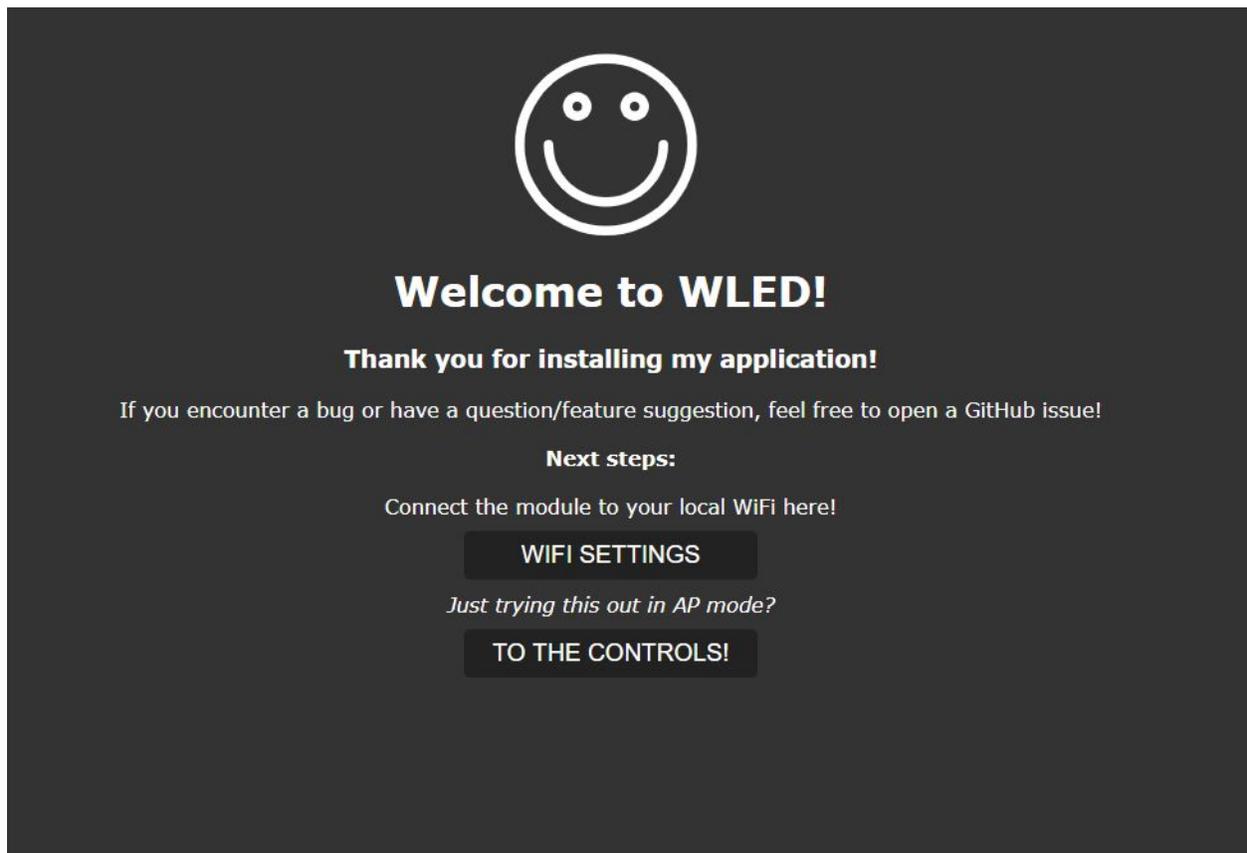
Step #5 - Upload the WLED firmware to your ESPixelPOPs

- Launch the ESPHome-Flasher application again
- Ensure the Serial Port dropdown box has the correct port selected for your USB-TTL adapter (You can refresh the list of available serial port devices using the button to the right of the drop down).
- Select the desired firmware file using the Browse button.
- Power cycle the ESPixelPOP but this time hold down the push button on the ESPixelPOP PCB and release after a few seconds to put it into the programming mode.
- Press the “Flash ESP” button on the ESPHome-Flasher application
- You should see the file transfer and programming progressing in the “Console” window.
- You will see a “Done! Flashing is complete!” message when done
- Power cycle your ESPixelPOPs



Step #5 - Connect and configure WLED for your network

- Using your laptop check for a new Wi-Fi Access Point SSID called `WLED-AP`. If you don't see it try refreshing your AP list (on Windows 10 I turn off the Wif-Fi , then turn it back on).
- Connect your laptop to the WLED SSID using the password: **wled1234**
- Fire up your browser and it should get automatically redirected to the WLED web page
- If not then point your browser to <http://4.3.2.1> and you should see this page.



- Click on the WIFI SETTINGS button
- Enter the following information as required by your Wi-Fi network
- Network Name: (Your home Wi-Fi SSID)
- Network Password: (Your home Wi-Fi password)
- Static IP : (Desired IP for your home network)
- Static Gateway: (Home network gateway IP)
- Static subnet mask: (Home network subnet mask, typically 255.255.255.0)
- When done hit the Save & Connect button

Back

Save & Connect

WiFi setup

Connect to existing network

Network name (SSID, empty to not connect):

Your_Network

Network password:

Static IP (leave at 0.0.0.0 for DHCP):

0 . 0 . 0 . 0

Static gateway:

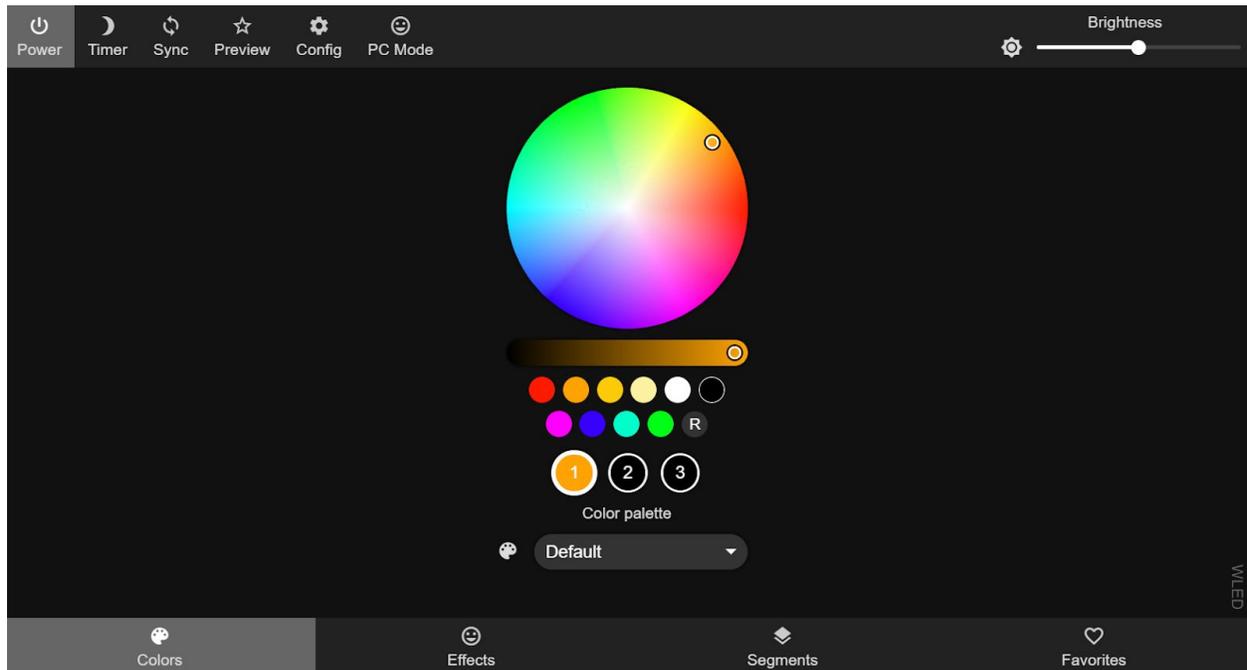
0 ▾ . 0 . 0 . 0

Static subnet mask:

255 . 255 . 255 . 0

Step #6 - Setup your Pixels on WLED

- Reconnect your laptop to your home Wi-Fi network
- Point your browser to the IP you configured WLED for
- You will get the main WLED page



Now go to the Config tab and then the LED Preferences tab. There are a lot of options on this page, we are only going to configure what is needed to get going.

LED count: (Set this for how many pixels you have connected)

WLED will inform you of what sized power supply is required for the configured pixel count

Enable automatic brightness limiter: (Enable this to allow software limiting of current)

Maximum Current: (Configure this for the power supply you have connected)

Enabling this ensures the software will not try to make the LEDs any brighter than you power supply can handle *BUT* it doesn't protect you from a software (or hardware) glitch doing so.

LEDs are 4-channel type(RGBW): (Enable this if you have RGBW pixels connected)

Color Order: (Set this for the pixels you are using)

Step #7 - Going Forward

I have just touched on the basics of getting WLED running on your ESPixelPOPs. For more information of the features you can refer to the WLED wiki:

<https://github.com/Aircoookie/WLED/wiki>

Another good site is the WLED Discourse site:

<https://wled.discourse.group/>

USB-TTL Adapter Programming Notes:

Do not use your USB-TTL adapter to power the ESPixelPOP during programming as it cannot supply enough current to reliably power the ESP-01 module. Make sure you power the ESPixelPOP board from a separate source that can supply at least 300 mA.

Only connect the USB -TTL adapter using the GND, RX & TX connections with jumper wires. If your USB-TTL adapter has sockets do not connect it directly to the pins on the ESPixelPOPs board. Do not connect any of the other pins from the USB-TTL adapter.

You can use either a 3.3V or a 5V version of a USB-TTL adapter. The diode in the ESPixelPOPs circuit acts as a level translator so the ESP-01 only sees the correct voltage either way.

I have found some cheap USB-TTL adapters do not seem to work as well as others for programming the ESP-01 module on the ESPixelPOP board. If you are having issues I recommend you try a different adapter. If you still have issues you may want to make or purchase a dedicated ESP-01 programmer.