

# ESPixelStick Assembly and Usage

## OVERVIEW

The ESPixelStick is a wireless (802.11g/n) pixel controller that interfaces as a standard E1.31 controller and supports a variety of pixel types. The firmware is open source and written in the EPS8266 Arduino environment to allow the user to modify and experiment with the firmware as they see fit. The ESPixelStick is provided in kit form with all SMD components pre-soldered. It is a fairly easy kit to solder with a low component count.

## KIT CONTENTS

For each ESPixelStick, your kit will contain the following:

Quantity	Part Description
1	ESPixelStick Main Board (SMD pre-soldered)
1	100uF 50V Electrolytic Capacitor
2	100uF 16V Electrolytic Capacitor
2	6x6mm Tactile Switch
2	1x3 Male Header
1	2x4 Female Header
1	6 Pin Plug
1	6 Pin Right Angle Socket
1	Mini ATM Fuse Holder
1	3A Mini ATM Fuse
1	Jumper
1	*ESP-01 Module (optional)

*\*The ESP-01 Module is required for completion, but is optional when purchasing the kit.*

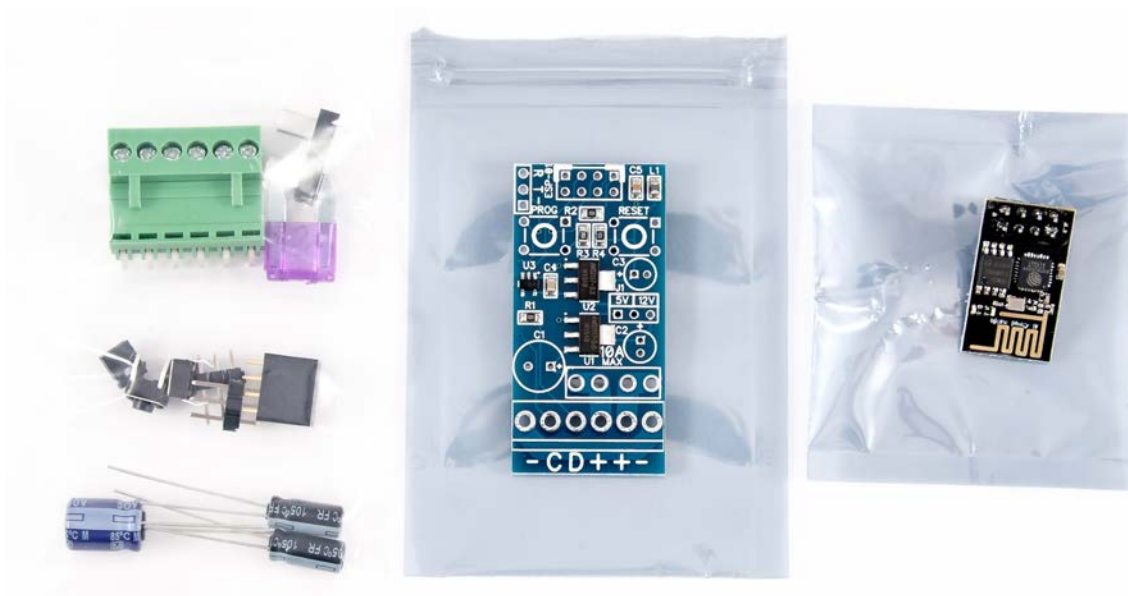


Figure 1 - ESPixelStick BOM

## ASSEMBLY INSTRUCTIONS

Before beginning assembly, it is recommended that you fully read the Assembly Instructions section. Verify your BOM (Bill of Materials / Kit Contents above) and visually inspect the ESPixelStick Main Board. Verify that there are no issues with the board such as broken traces or shorted components. The ESPixelStick Main Board has already been visually inspected, but it's always recommended to perform a final check before assembly. Many of the reference descriptors were removed from the board due to space constraints, so some component locations are referred to graphically or by function. The order of assembly really isn't that important, but the following is recommended.

- Install and solder **6x6mm Tactile Switch**, at **PROG** and **RESET**.
- Install and solder **2x4 Female Header**, at **ESP-01**.
- Install and solder **Mini ATM Fuse Holder**, below where it says "**10A MAX**".
- Install and solder **1x3 Male Header**, at **J1**. This is the operating voltage selection jumper.
- Install and solder **1x3 Male Header**, next to **ESP-01** and **PROG**. This is the programming header.
- Install and solder **6 Pin Right Angle Socket** at the end of the board, below the fuse holder.
- Install and solder **100uF 50V Capacitor**, at **C1**. Note the orientation of the positive and negative as this component is polarized and must be installed correctly.
- Install and solder **100uF 16V Capacitor**, at **C2** and **C3**. Note the orientation of the positive and negative as this component is polarized and must be installed correctly.
- Trim the leads of **C1**, **C2**, and **C3** on the bottom of the board.
- Insert the **ESP-01 Module** into the 2x8 Header, at **ESP-01**. The module must be inserted with the antenna portion (gold zig-zag) oriented away from the board.
- This completes assembly of the ESPixelStick.

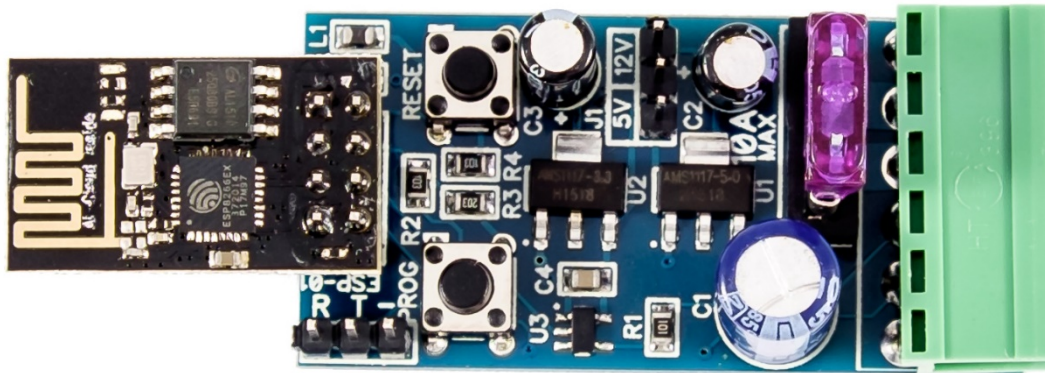


Figure 2 - Completed ESPixelStick

## HARDWARE CONFIGURATION

There is only one hardware configuration item, Jumper **J1**. However, it is **CRITICAL** that this is set correctly or you will **DESTROY** your ESPixelStick (more specifically, component **U3** – the level converter).

**\*\*\* WARNING \*\*\***

**If you connect to a 12V supply with J1 set to 5V, YOU WILL LET THE MAGIC SMOKE OUT!!!**

**\*\*\* WARNING \*\*\***

Place the **Jumper** on **J1** in either the **12V** or **5V** location, depending upon your operating voltage. Again, supplying **12V** with **J1** set to **5V** will kill your ESPixelStick.

All power and pixel connections are made through the 6 Pin connector located on the end of the board. Wiring designators are indicated on the bottom of the ESPixelStick. The ESPixelStick is designed to be located near the element that it's controlling, eliminating the issue of having to deal with long pixel transmission lines and null pixels.

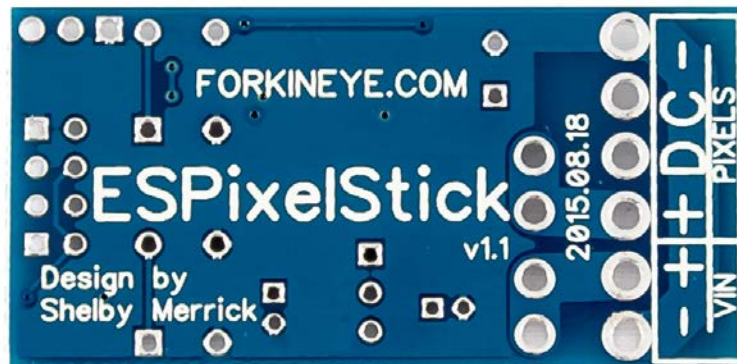


Figure 3 - Bottom of ESPixelStick

- **VIN -** : Power Supply Ground
- **VIN +** : Power Supply Positive
- **Pixels +** : Pixel Positive
- **Pixels D** : Pixel Data Line
- **\*Pixels C** : Pixel Clock Line
- **Pixels -** : Pixel Ground

\*Note that clock-less 3-wire pixels (WS2811/2, GECE, etc...) will not use the **Pixels C** connection. It is reserved for future support of clocked pixels such as WS2801.

Usage of the onboard fusing and power distribution of the ESPixelStick is optional. If your pixels are being fed power directly by a separate power supply or fuse block, the ESPixelStick will be back-fed power by the pixel power connection (**Pixels +** and **Pixels -**).

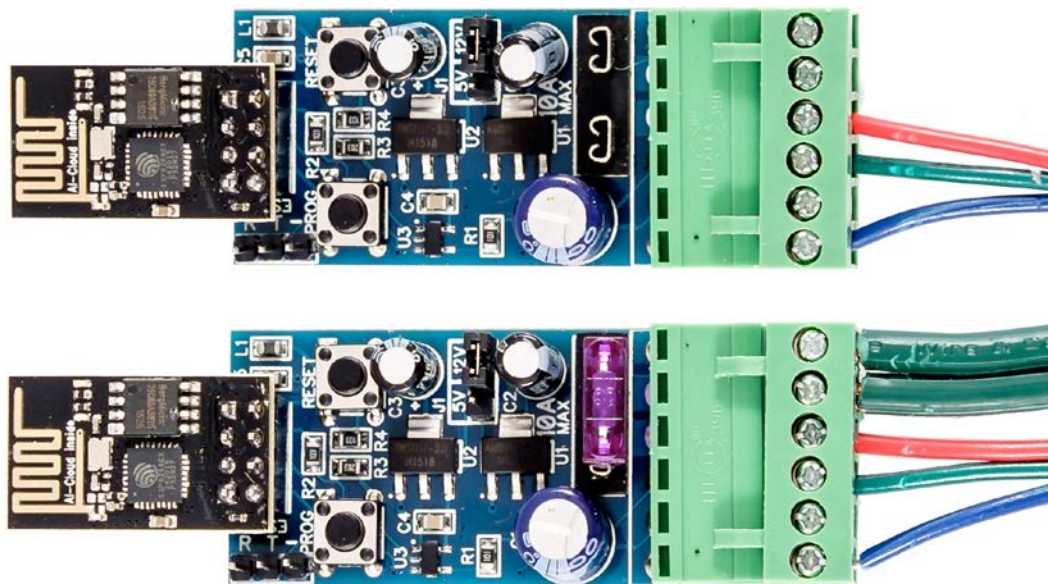


Figure 4 – Back fed power (top) and power fed through ESPixelStick (bottom)

## PROGRAMMING

Programming is done in the Arduino environment and requires a suitable USB to Serial adapter with the GND, TX, and RX lines exposed. Such adapters are commonly referred to as “USB to TTL” and will advertise serial chips such as FTDI, PL2303, CP2102, CH340G. CP2102 based devices are recommended and they can be bought for just a few dollars online. If utilizing a FTDI programmer, you must ensure that is configurable for 3.3V operation or you may experience programming issues. We’ll go over some requirements, setting up the Arduino environment to support the ESP8266, connecting the ESPixelStick and finally programming the ESPixelStick.

## REQUIREMENTS

- USB to TTL Serial Cable
- Arduino 1.6.5 - <http://www.arduino.cc/en/Main/Software>
- Latest ESPixelStick Firmware - <http://github.com/forkineye/ESPixelStick/releases>

## SETTING UP THE ESP8266 ARDUINO ENVIRONMENT

- Install Arduino 1.6.5 - <http://www.arduino.cc/en/Main/Software>
- Start Arduino and open the *Preferences* window by going to **File->Preferences**.
  - o In the *Additional Boards Manager URLs* field, enter the following:  
**`http://arduino.esp8266.com/stable/package_esp8266com_index.json`**
  - o Click **OK**.
- Open the *Boards Manager* window by going to **Tools->Board->Boards Manager**.
  - o Scroll to the bottom, and click on the **esp8266** row.
  - o Click **Install**.
  - o The ESP8266 download is ~150MB and may take a few minutes to install.
  - o Click **Close**.
- Select the appropriate board by clicking on **Tools->Board->Generic ESP8266 Module**.
- Select the appropriate CPU clock by clicking on **Tools->CPU Frequency->160 MHz**.
- The Arduino environment is now configured to program your ESPixelStick.

## PROGRAMMING INTERFACE

The ESPixelStick must be connected to a power supply through the main plug for programming. From the programming connector, connect the **R**, **T**, and **-** lines to the **RX**, **TX**, and **GND** connections of your USB to TTL Serial Cable.

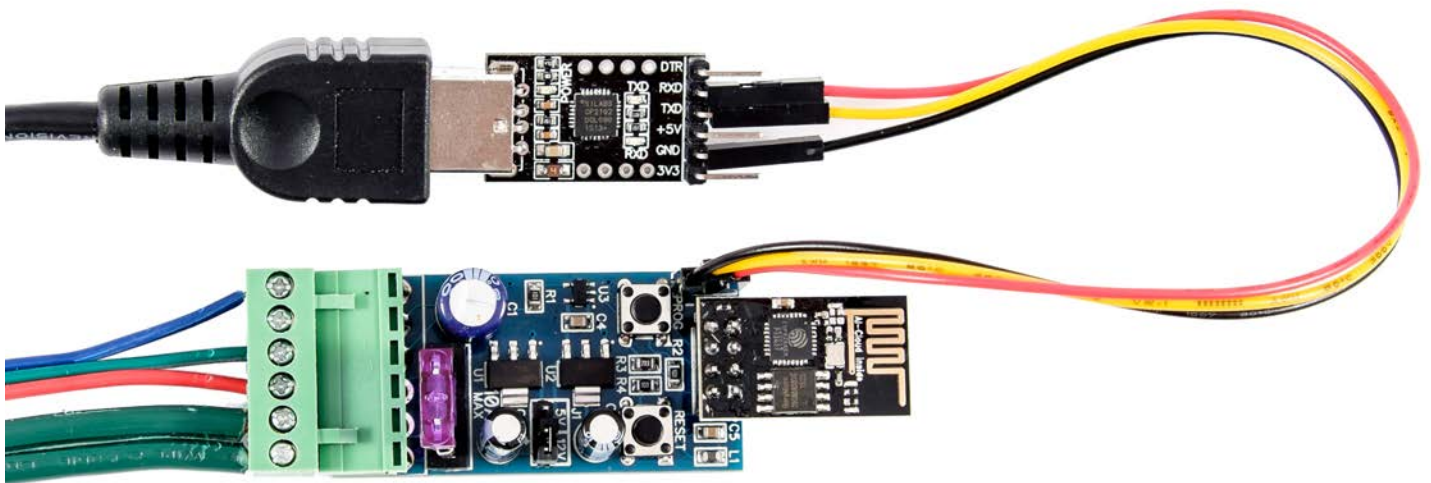


Figure 5 - Programming Connections

Tip – If programming multiple controllers, you may find it easier to dedicate one controller as the programmer until done. All code and configuration is stored on the ESP-01 module itself, so you can just swap out modules on the ESPixelStick Main Board that you're using as the programmer. Note however, it is recommended that you disconnect the power to the ESPixelStick while swapping modules. The programming cables however may stay connected.

- Download the latest ESPixelStick Firmware release from <http://github.com/forkineye/ESPixelStick/releases>.
- Extract the ESPixelStick release archive into its own directory within your Arduino Sketchbook directory, typically **C:\Users\ on a Windows machine.
 
  - If you are unsure of where your Sketchbook location is, you can check within the Arduino IDE by clicking on **File->Preferences**.
  - Note – Do not put the ESPixelStick firmware in your Arduino Library directory or it will not compile correctly. It **MUST** be in your Sketchbook directory.**
- Rename the directory you extracted to **ESPixelStick**. It will be **ESPixelStick-1.x**.
- Open the ESPixelStick sketch by clicking on **File->Open** and navigating to the **ESPixelStick** directory.
- Double-click on **ESPixelStick.ino**.
- Scroll down to the section, **/\* BEGIN – User Configuration Defaults \*/**.
- Edit the fields as documented in the sketch. Only **ssid** and **passphrase** are required.
  - The other fields are optional defaults and can be set now or later from the Web GUI.
- Verify the correct serial is configured by checking **Tools->Port**.
- Open the *Serial Monitor* by clicking on **Tools->Serial Monitor**.
- In the bottom right of the *Serial Monitor* window, set your baud rate to **115200 baud**.
- Verify your ESPixelStick is connected and powered as detailed in the **Programming Interface** section above.
- Enter programming mode on the ESPixelStick with the following procedure
  - Hold down the **PROG** button.
  - Press and release the **RESET** button.
  - Wait 2 seconds.
  - Release the **PROG** button.
  - The ESPixelStick is now in programming mode.
- Compile and upload the firmware by clicking on **Sketch->Upload**.
- Once the upload is complete, the ESPixelStick will relay its configuration status to the *Serial Monitor* window. You can now connect to the ESPixelStick with your web browser for further configuration if required. It's a good idea to record the IP Address as chances are, you'll need it!
- Your ESPixelStick is now programmed.

## CONFIGURATION AND USAGE

Configuration and usage of the ESPixelStick is pretty straight forward for anyone familiar with 802.11 wireless networks and E1.31. Here are some details to get you up and running.

### A NOTE ON 802.11 WIRELESS NETWORKS FOR SHOW USAGE

Just like a wired show network, your wireless show network should be separate from your home network and dedicated solely for your show. If you're planning to use Multicast, be aware of the issues that may be caused by non-show devices attempting to access your network and how to mitigate them. Mainly, keeping all non-show device off the network and disabling SSID broadcast for your show AP. If you must use your home network, Unicast only is recommended and be aware that other devices on your network may degrade your show performance.

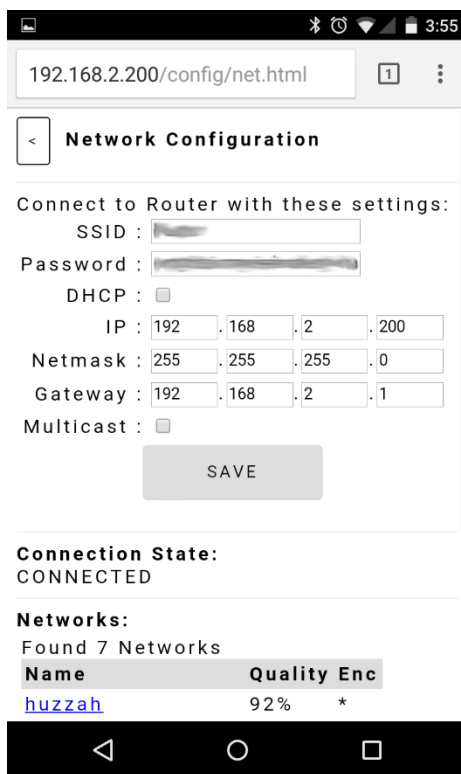
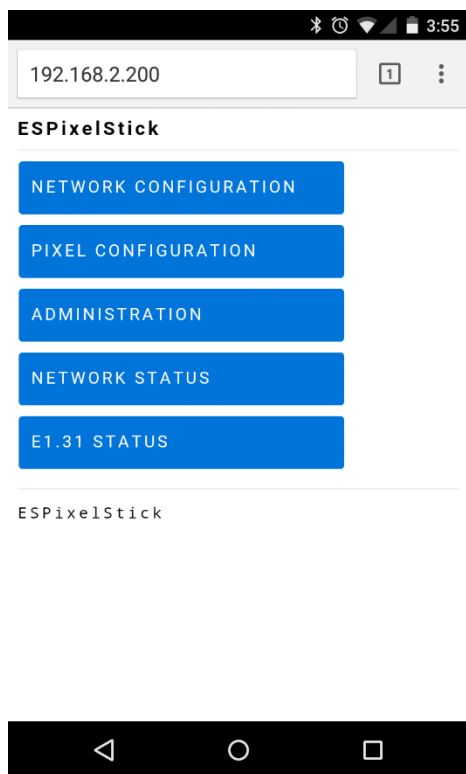
### WEB BASED ADMINISTRATION

The ESPixelStick has a mobile friendly web-based administration interface that provides configuration and status reporting. To connect, simply browse to the IP Address reported by your ESPixelStick during programming.

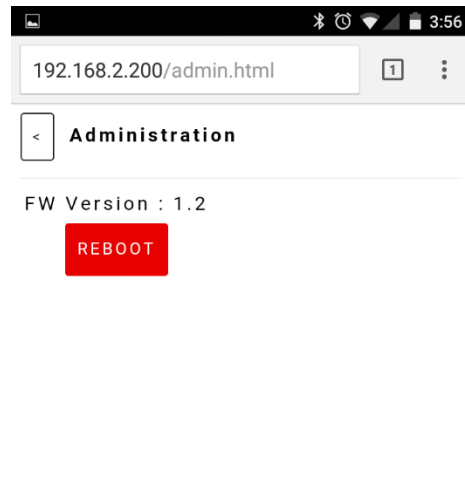
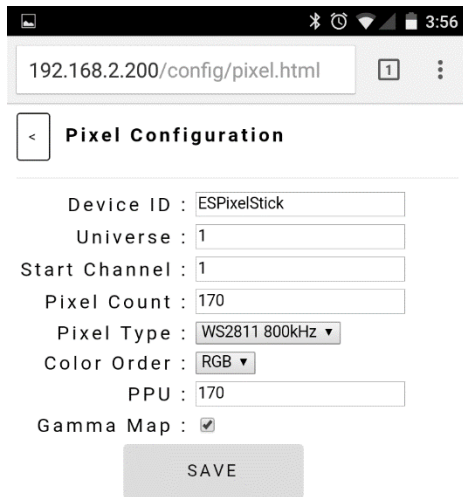
#### \*\*\* NOTE \*\*\*

It is currently not recommended to use the Web Based Administration interface while show data is streaming to the ESPixelStick. Doing so consumes many resources and may cause issues in the ESP8266 core, causing the UDP socket responsible for listening to show data to crash, therefore requiring a reboot of your ESPixelStick. Not something you really want at show time! Fixes are being investigated for a future firmware release. Note that some sequencers will stream E1.31 data even while idle, such as Vixen 3.

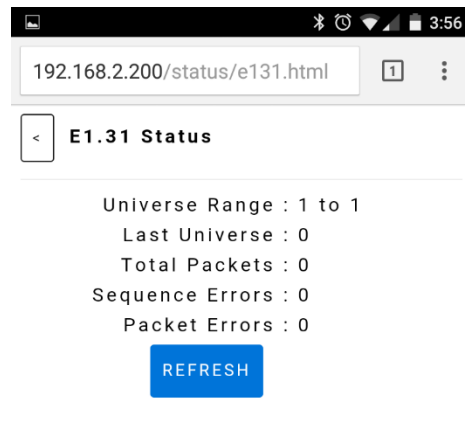
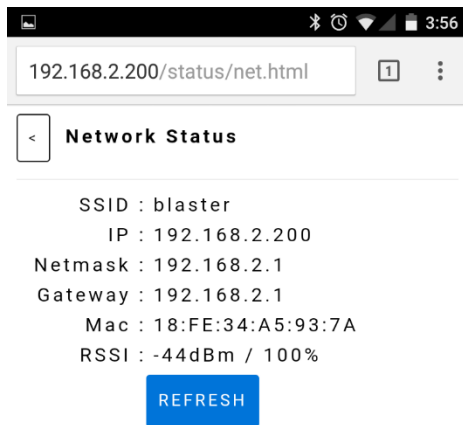
#### \*\*\* NOTE \*\*\*



The web interface home page and Network Configuration page are self-explanatory. Not however, **Multicast** is only supported for a single universe at this time.



The **Device ID** is a free-form field to help you know which ESPixelStick you are logged into it. The text here is displayed at the bottom of the homepage and in the title bar of your web browser. **PPU** (Pixels per Universe) controls how many pixels the ESPixelStick will expect per Universe. Most users will want to leave this to the default maximum of 170. **Gamma Map** applies only to 8 bit pixels and will apply a pseudo-gamma correction of 2.2. You may want to use this if you don't already have custom dimming curves setup in your sequencer.



The Network Status and E1.31 Status pages are self-explanatory.

## SEQUENCER CONFIGURATION

Directions will vary for all the various sequencing programs available. Follow your product's instructions for setting up an "E1.31" or "sACN (Streaming ACN)" controller.