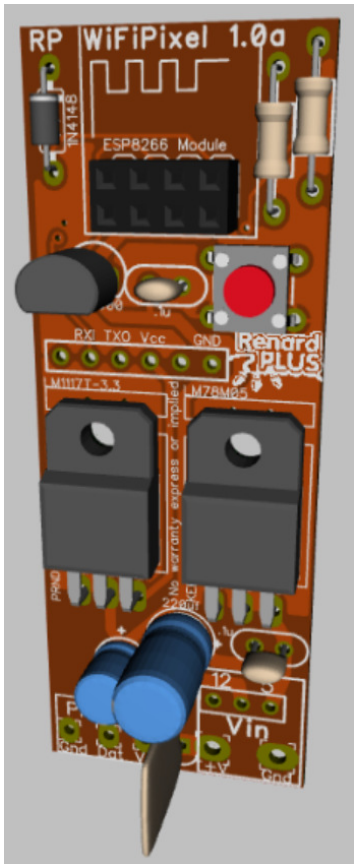
WiFiPixel

May 2016
Board Version 1.0a
Document Rev 1.2



Renard-Plus, Salem, Oregon 97302

Published 2016. Printed in United States

We wish to also thank the Do It Yourself Community for the inspiration it has given us in the development of this product.

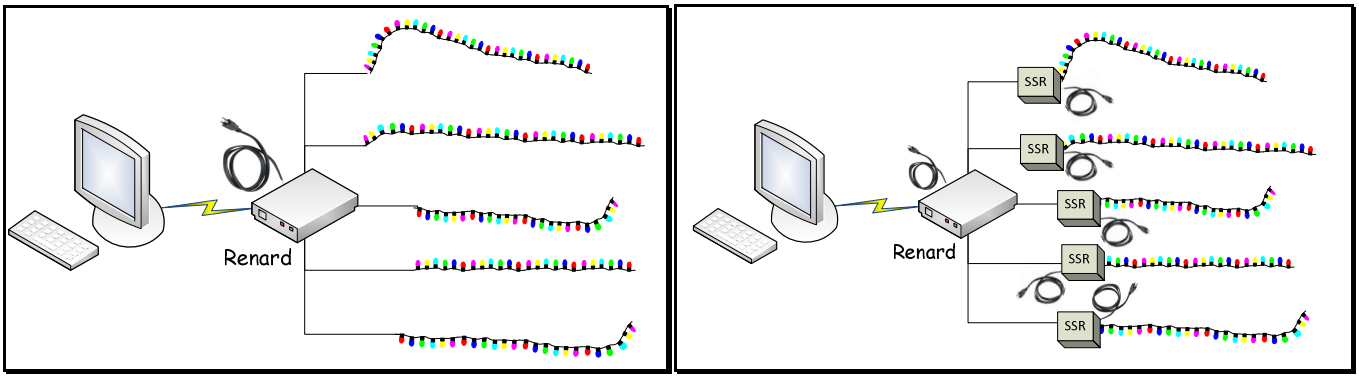
Table of Contents

1. Introduction to Renard

Renard is the name of a “do-it-yourself” (DIY), computer-controlled, PIC-based dimmer light control concept. It also refers to a family of dimming controllers that have been designed and built based on this concept.

The Renard design concept was originally described by Phil Short in the Simple PIC-Based 8-Port Dimmer 'How-To' on the http://computerchristmas.com website. Since then there have been many enhancements and new designs based on this hardware. There have been many contributors to advancing Renard technology including M. Macmillan, D. Davis, P. Rogers, T. Straub, D. Haberle, A. Williams and others

Renard controllers typically rely on a separate computer running a light sequencing program to send it real-time sequences of controller commands to sequence the lights. The computer communicates with the Renard via RS232, RS485, or wireless (depending on the design) and the Renard controls the lights either through built-in power control (power is output directly to the lights), or via separate “SSRs” (solid state relays supply the power when commanded by the controller).



Example Renard configurations

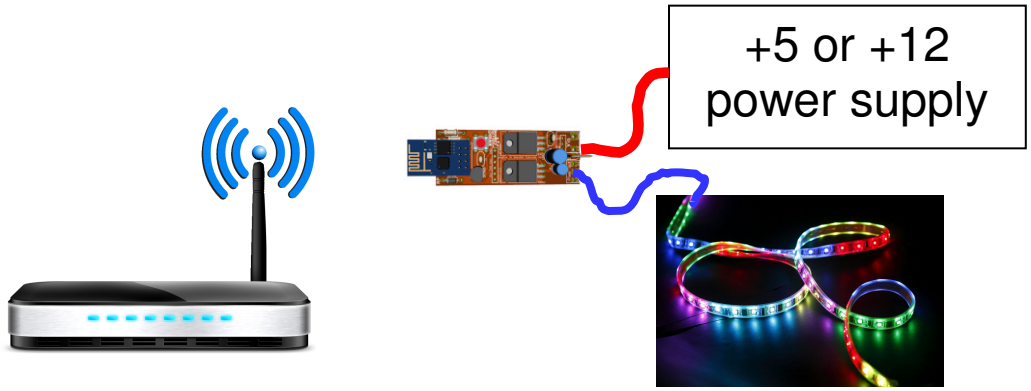
Output of the Renard can be control signals (to an SSR), direct AC line voltage (110, 100/220, or 220), DC voltage, pixel data or a combination of these depending on the design.

Renard is a DIY hobbyist effort and there is a vast amount of products and related peripherals to select from including the Renard Plus ATX Adapter. To obtain a specific design, there might be “buy a parts kit and/or blank PCB” offering at a site (such as from www.renard-plus.com or www.renard-shop.com), “etch it yourself” files for true DIY, or coop/group buys for kits and PCBs also in forums (like DIYChristmas.org).

2. Overview of Renard Plus WiFiPixel Adapter

Renard Plus controller boards are flexible in their input configuration and can support a wide range of interconnect options including wired and wireless. For wireless connections, there are a number of communications types that can be used including XBee, NRF, and standard 802.11 “Wi-Fi” to any number of LED control devices.

The RP_WiFiPixel, as the name implies, is a Wi-Fi in, pixel out controller. Even though the WiFiPixel controller does not actually utilize the Renard data format, it still fills a need in the Renard Plus family of controllers. It utilizes the incredibly capable and affordable ESP8266 ESP-01 wireless module as the “brains” of the adapter. The WiFiPixel board can be configured to be run from either 5v or 12v to drive either correspondingly 5v or 12v pixels (individual or strips). The board is designed for pixels that utilize a single data line such as the popular WS2811 / 2812 pixels.



Pixels are RGB LEDs that are considered “smart” as they typically have power, data in, and data out to allow them to be daisy chained so one controller can control many pixels. They are individually addressable so if you had a strip of 50 pixels, you can make each pixel be a different color or just turn on one pixel to any color anywhere on the strip. The options are nearly limitless and allows for some very striking light animation effects.

The Wi-Fi capability of the ESP8266 enables the WiFiPixel module to run on a standard wifi network.

The ESP module is programmed using the Arduino IDE found here:
https://www.arduino.cc/en/Main/Software

with the ESP support added per Adafruit’s guide here:
https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/using-arduino-ide

using the sketch from madsci1016 (Bill Porter) here:
https://github.com/madsci1016/ESPixelStick
or from forkineye (Shelby Merrick) here:
https://github.com/forkineye/ESPixelStick

Once the ESP has the pixel code in it, access to the pixels is handled through an on-board web server, and light animation data is sent using networking protocol sACN / E1.31 DMX. Your animation software must support E1.31 capability to work with the WiFiPixel. The Vixen animation software is a great example as it has such a capability either built-in or through an add-on.

Feature	Detail
Name	**Renard Plus WiFiPixel Adapter**
Target use	Wi-Fi to pixel adapter
Channel Count	512 DMX color channels = 170 RGB pixels
Input	5v or 12v (selectable), Wi-Fi
Output	Pixel data (ws2811/2812 protocol default)
Dimmable?	Pixels are dimmable via software
Status Indicators?	No
Channel Indicators?	n/a
Control Input – Renard	E1.31 sACN
Control Input – DMX	Yes native via E1.31
Daisy-chain output	n/a
Wireless	YES
On board programming	YES via USB to serial TTL adapter
Enclosure	1” Thin wall PVC Pipe
Heatsink?	Built-in
Distance?	Wi-Fi b/g/n distance
Speed?	Pixel max data rates.

3. Assembly Instructions

This section covers the construction of the Renard Plus WiFiPixel Adapter board. It approaches the assembly tasks as a learning exercise for new builders, so that they can develop proficiency and self-confidence. The project itself is quite simple and if you follow the steps *carefully*, you should have a working adapter when you are done. Additional information and guides on techniques and tools can be found in the "Tools and Parts ID Guide" at: www.renard-plus.com/files/Tools_and_Parts_ID_Guide.pdf

3.1 BOM / Parts List

The following is the Bill Of Material for building this board. The link to the Mouser project is:

http://www.mouser.com/ProjectManager/ProjectDetail.aspx?AccessID=tbd

Note: If you did not obtain a complete parts kit, Mouser is the most convenient place to order your needed parts. However, Mouser is not always the most cost effective source for parts- you may want to check alternatives like Tayda Electronics, DealExtreme, EBay, or other sources.

Picture	Description	Qty	Location	Mouser P/N	Other Source
	330 ohm resistor 1/8 watt Org-Org-Brn	2	330	299-330-RC	http://www.taydaelectronics.com/resistors/1-4w-carbon-film-resistors/10-x-resistor-330-ohm-1-4w-5-carbon-film-pkg-of-10.html
	100nf / .1uf cap	2	.1u	81-RDER71H104K0S1C03	http://www.taydaelectronics.com/capacitors/monolithic-ceramic-capacitor/0-1uf-50v-multilayer-ceramic-capacitor.html
	220uf 10v 5x11mm Electrolytic	2	220UF	598-227CKS010M	http://www.taydaelectronics.com/capacitors/electrolytic-capacitors/220uf-10v-105c-radial-electrolytic-capacitor-5x11mm.html
	1N4148 diode	1	1N4148	583-1N4148-T	http://www.taydaelectronics.com/diodes/standard/1n4148-switching-signal-diode.html
	Polyswitch fuse 30v 3A (Optional- jumper location if not used)	1	PTC	650-RUEF300-1	http://www.taydaelectronics.com/circuit-protection/polyswitch-devices/poly-switch-resettable-ruef-series-30v-3a.html
	2N7000 N-Channel MOSFET	1	2N7000	821-TSM2N7000KC821-T	http://www.taydaelectronics.com/t-transistors/2n-series/2n7000-mosfet-n-channel-60v-0-2a.html
	LM78M05 5 volt regulator (Optional if 5V in with 5V pixels are used)	1	LM78M05	863-MC78M05CTG	http://www.taydaelectronics.com/lm7805-l7805-7805-voltage-regulator-ic-5v-1-5a.html
	LM1117T-3.3 LDO 3.3 volt regulator	1	LM1117T-3.3	926-LM1117T-3.3/NOPB	http://www.taydaelectronics.com/ic-integrated-circuits/voltage-regulators/lm1117t-lm1117-low-dropout-voltage-regulator-ic-3-3v.html
	tactile switch 6*6mm through hole SPST – NO	1	Switch	706-95CAP-011	http://www.taydaelectronics.com/tact-switch-6-6mm-7mm-through-hole-spst-no-3752.html
	2x4 2.54mm Female Header	1	ESP8266 Module	855-M20-7830446	http://www.taydaelectronics.com/connectors-sockets/pin-headers/2x20-pin-2-54mm-double-row-female-pin-header.html Cut off to 2x4 with Dremel, coping saw, or snips.

Picture	Description	Qty	Location	Mouser P/N	Other Source
	ESP8266 ESP-01 module	1	ESP8266 Module	n/a	http://www.ebay.com/itm/ESP8266-Esp-01-WiFi-Wireless-Transceiver-Module-Send-Receive-LWIP-AP-STA-/322028163125
	1x9 (or more) Male Header cut to fit (optional)	1 opt	Vin voltage selection, & Prog. Port	571-1-826646-6	http://www.taydaelectronics.com/connectors-sockets/pin-headers/40-pin-2-54-mm-single-row-pin-header-strip.html
	4-40 bolt, nut and lock washer (optional)	2 opt	LM78M05, LM1117-3.3	534-9900	Home Depot, Ace Hardware, local hardware store.
	USB to Serial TTL CP2102	1	Prog Port	n/a	http://digwdf.com/store/product.php?id_product=178 http://www.ebay.com/itm/262213765221 Note, this one can be used to program Arduinos and plugs straight into the programming port as well.

3.2 Case

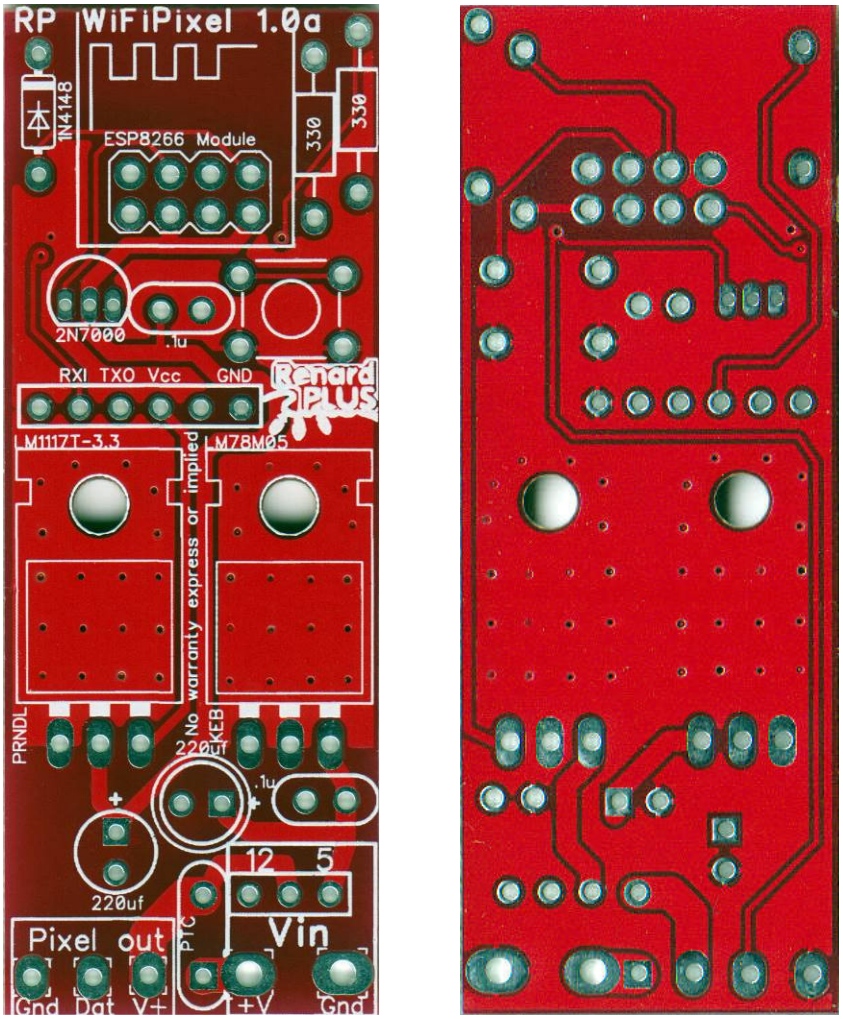
The WiFiPixel is designed to be mounted in 1” thin wall PVC pipe (or Grey PVC conduit works) with either one or two caps to provide a weather resistant case. When using one cap, mount with the cap up and stuff the bottom opening (where the wires come out) with foam to keep water splashes out.

3.3 Assembly

This board is a very simple device to assemble and test. It is easiest if you follow these instructions, checking off steps as they are performed. This will lead you through the assembly installing components from shortest/smallest to tallest.

3.3.1 First Things First

1. Begin by inspecting the PCBs to look for any defects such as cracks or breaks. The holes on the board should be open on both sides



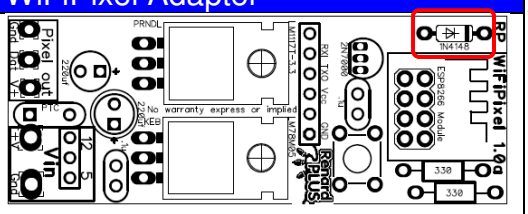
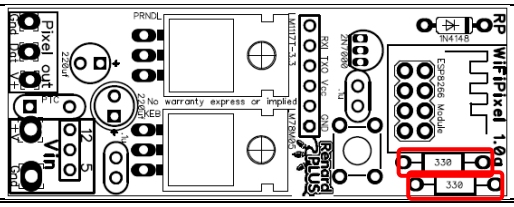
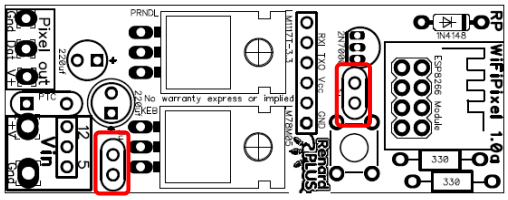
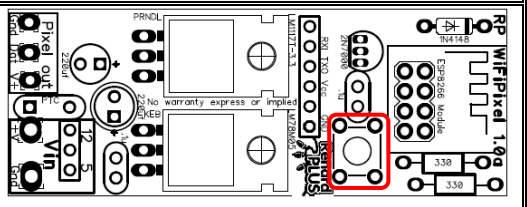
2. Next inspect and sort out the various parts for the board. Make sure you understand which parts are which, and things like resistor codes and component orientation. A separate document on these concepts is available at
   www.renard-plus.com/files/Tools_and_Parts_ID_Guide.pdf
   and on other resource sites like Wikipedia.
3. Follow the assembly guide as follows in the next section.

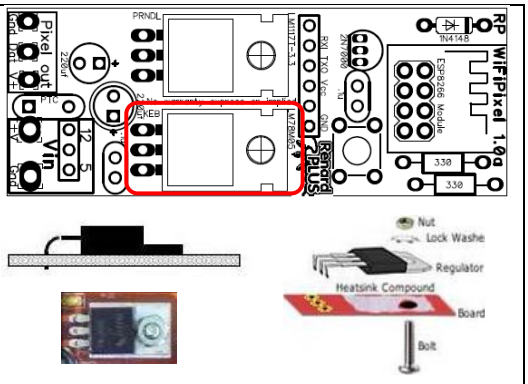
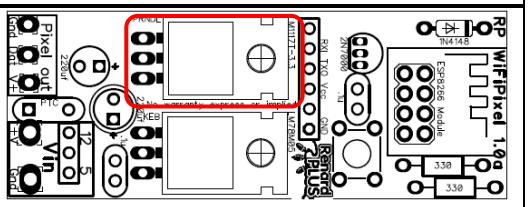
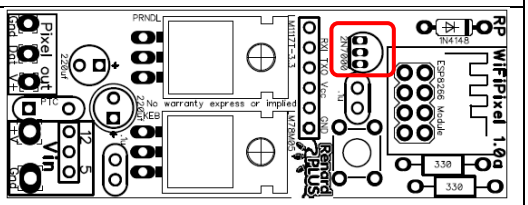
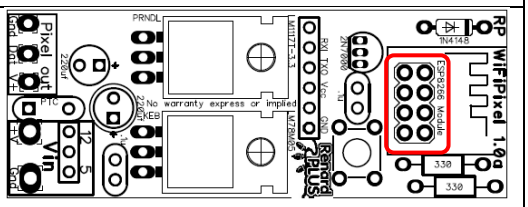
3.4 Parts Installation

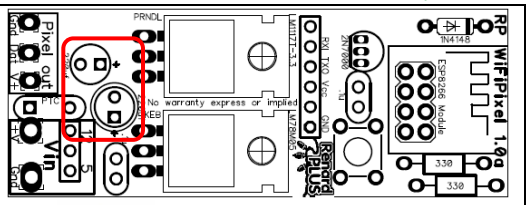
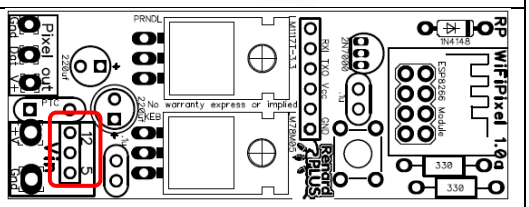
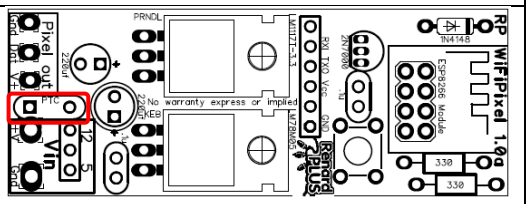
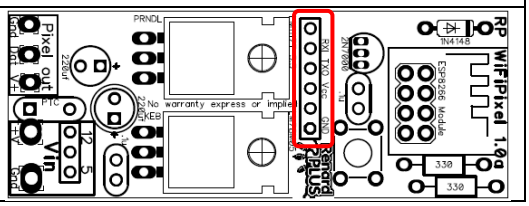
Generally, the following component assembly order is grouped from shortest to tallest parts to make assembly easier. Special instructions for component orientation should be listed if a component has any. Don't stress it- we try to make this as easy as possible!

3.4.1 Step-by-step

If a component has a required orientation, we will be sure to mention it in the Instructions section for the component. Things like resistors do not have a specific orientation and can be installed either direction. The VALUE is important and that is indicated by the colored strips or printed on it. Parts like LEDs, diodes, and electrolytic caps DO have a specific orientation and the Instructions will indicate so. See the Tools and Parts ID document on www.renard-plus.com for details.

Step	Instructions	WiFiPixel Adapter
1 ☐	Install the 1N4148 diode at the location marked "1N4148". Solder and clip leads. *Diodes have a specific orientation. The stripe on the diode must match the stripe on the silkscreen. Stripe to right as shown.*	
2 ☐	Install the two 330 ohm (Orange Orange Brown) at the locations marked "330". Solder and clip leads *Resistors do not have a specific orientation.*	
3 ☐	Install the .1uf (marked 104) capacitors at the two locations marked ".1u". Solder and clip leads. *Bypass caps do NOT have a specific orientation.*	
4 ☐	Install the Tactile Switch at the switch location indicated on the silkscreen. Solder *The switch can be installed in either of the two orientation where the pins line up with the through holes.*	

5 ☐	If you will (or might) use the board with 12V supply voltage to drive 12V pixels, install the LM78M05 (or LM7805) at the location marked “LM78M05”, forming the leads as indicated in the drawings to the right. Fold the pins over the shaft of a small screwdriver to create smooth bends. Apply an even layer of heat sync compound on the back of the regulator (or use a silicon thermal pad) and insert the leads into the proper holes. Secure it with a 4-40 x1/4” or M3 x4mm bolt, #4 lock-washer, and 4-40 or M3 nut. Solder and clip leads. *Part has a specific orientation as shown!*	
6 ☐	Install the 3.3v LDO regulator LM1117-3.3 at location marked “LM11117-3.3”, forming the leads as indicated. Fold the pins over the shaft of a small screwdriver to create smooth bends. Solder and clip leads. Bend up the regulator a little so it is not contacting the board or follow the OPTIONAL steps below. *Part has a specific orientation as shown!* <u>Optional</u> – Place a to-220 silicone thermal pad (Sil-Pad) under the regulator and secure it with a **<u>NYLON / plastic</u>** 4-40 or M3 bolt, #4 lock-washer, and 4-40 or M3 nut. A metal bolt MAY short out the 3.3v on the tab.	
7 ☐	Install the 2N7000 transistor (TO-92) at the location marked “2N7000”. It is best installed close to the board without it touching- maybe a millimeter above the board. Solder, clip leads, and **check for solder bridges!** *Part has a specific orientation – match the flat side of the transistor to the flat side shown on the silkscreen.*	
8 ☐	**<u>Optional (but recommended)</u> -** Install the 2x4 Female Header / socket for the ESP8266 Module at the location marked “ESP8266 Module”. *If you will not be using a socket for the wireless module, leave this spot open until AFTER voltage testing that follows assembly.*	

9 ☐	Install the two 220uf electrolytic caps in the spots marked “220u”. If one of your caps is smaller, please mount it in the upper location closest to the edge of the board. *Be sure that the +lead is installed in the hole marked with a “+” symbol. The +lead is usually longer than the -lead, and the -lead is usually identified by a black or white stripe.*	 
10 ☐	Using a clipped lead from the resistors or diodes, bend it and place it in the center hole and either 12V or 5V location to more securely set the input voltage to either 12v or 5V. ***If you apply 12v to the WiFiPixel when the jumper is set for 5v, it will more than likely smoke the board and/or any 5V pixels attached!!!*** <u>Optional</u> - install a 3 pin header and a jumper to allow “in field” changing of the setting if desired.	
11 ☐	Install the polyfuse at location marked “PTC”. Solder and clip leads *This part does not have a specific orientation.* <u>Optional</u> - It is possible to replace this part with a wire soldered across the connections but you then risk burning up the board if your string of pixels short out the power.	
12 ☐	Optional- Install a 6 Pin programming header at the indicated location. This is used to update the code on the ESP-01 module via a USB to Serial TTL adapter.	

3.4.2 Initial Testing

At this point you have completed the assembly of the board and you should gently clean the board of any residue and inspect for solder bridges or cold solder joints. What you are looking for are any solder bridges especially around the connectors and other closely spaced parts, or pins that are not fully and cleanly soldered. Look carefully for solder connections that did not flow to the pad properly- connections should be smooth and even, not “blobby”.

Next, you will want to check voltages to make sure the regulators are working, especially the 3.3v LM1117-3.3 as the ESP8266 MUST only ever be run on 3.3v. Hopefully you did not install the ESP8266 module yet, but if you did, hopefully it is in the socket and you can remove it. If you soldered it in, then hopefully you followed the other install instructions carefully or you could kill the ESP module when power is applied with other issues on the board. You should NOT have any pixels attached yet either!

3.4.2.1 5V Operation pre-check

If you built the board as a 5V only or have the Vin jumper set to 5V, then do the following:

- Make sure the LM1117-3.3 is in the proper location. If not, CORRECT IT before applying power.
- Make sure the 5V to center pin on Vin is the only pin set with a jumper. If not CORRECT IT before applying power!!
- Apply 5V to the Vin. + goes to the pad marked +V and – goes to the pad marked Gnd.
- Measure the outer two pins of the “Pixel Out” ground/- is marked Gnd, and the positive/+ is marked V+. You should see 5V here.
- Measure the voltage on the serial connector (RXI TXD Vcc GND) across the Vcc and GND pads. You should see 5v.
- Measure the voltage on the LM1117-3.3 middle pin (+) to a GND connection used before. You should see 3.3v
- Finally, measure the voltage on the ESP8266 connector on the end pin closest to the tactile switch location to any GND connection. You should measure 3.3v.

If everything is OK, turn off the power and move on to Final Assembly.

3.4.2.2 12V Operation pre-check

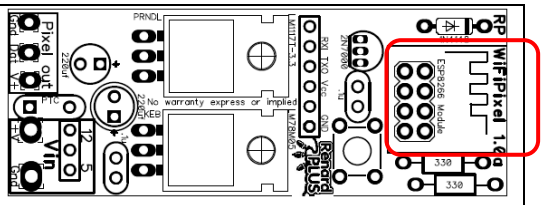
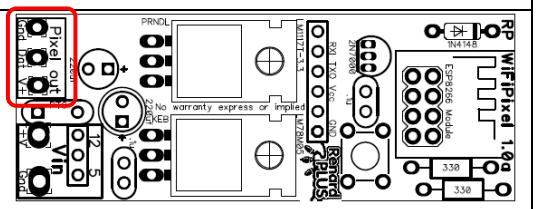
If you built the board as a 12V only or have the Vin jumper set to 12V, then do the following:

- Make sure the LM1117-3.3 is in the proper location. If not, CORRECT IT before applying power.
- Make sure the LM78M05 (or LM7805) is in the proper location. If not, CORRECT IT before applying power.
- Make sure the 12V to center pin on Vin is the only pin set with a jumper. If not CORRECT IT before applying power!!
- Apply 12V to the Vin.
- Measure the outer two pins of the “Pixel Out” ground/- is marked Gnd, and the positive/+ is marked V+. You should see 12V here.
- Measure the voltage on the serial connector (RXI TXD Vcc GND) across the Vcc and GND pads. You should see 5v (yes Five Volts!). 5V is used for data and to feed a low enough voltage to the 3.3v regulator so as to not blow it up.

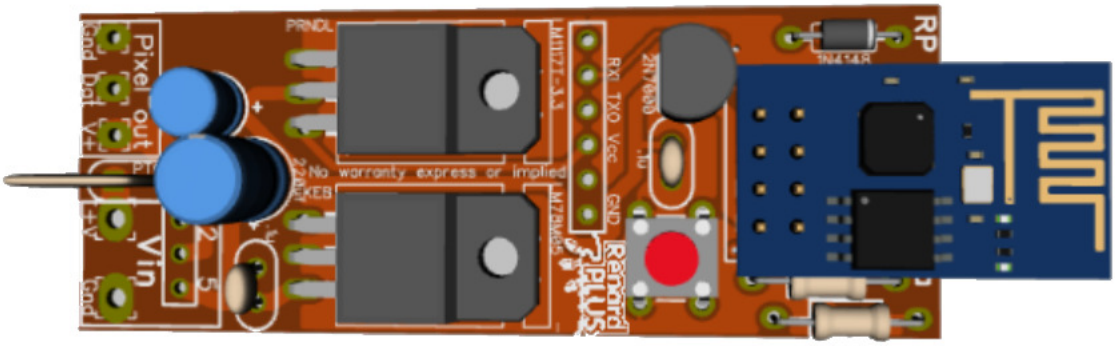
- Measure the voltage on the LM1117-3.3 middle pin (+) to a GND connection used before. You should see 3.3v
- Finally, measure the voltage on the ESP8266 connector on the end pin closest to the tactile switch location to any GND connection. You should measure 3.3v.

If everything is OK, turn off the power and move on to final assembly.

3.4.3 Final Assembly

12 ☐	Install the ESP8266 ESP-01 Module in the location marked “ESP8266 Module”. If you installed the female header as recommended, you can just plug it in to the socket. If you opted to omit the socket, place the wireless module into the pads on the board and solder. It will hang over the edge of the WiFIPixel board to allow good antenna operation.	
13 ☐	Attach the appropriate voltage 3 wire pixels (Typ WS2811 or 2812) to the Pixel Out location going to the Pixel IN end of the pixel string or bulb. You will need to refer to your particular pixel’s instructions or figure out which wires are which signals on the pixel (see following section on pixels). Usually it will be marked on the pixel itself, or you may need to look inside the head of the pixel to see where the wire attach to read the function order of the wires.	

3.4.4 Completed Board



4. Connections

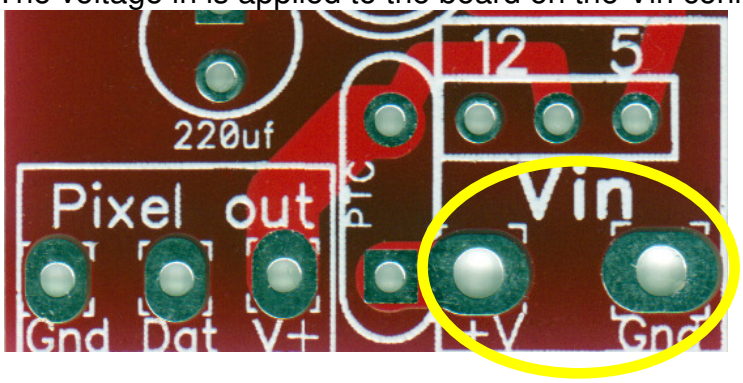
The WiFiPixel is very straightforward in operation. This section of this guide will attempt to describe the various connections both in and out of the board.

4.1 WiFiPixel I/O

The WiFiPixel takes in voltage (either 5v or 12v as determined by the pixel type you use) and a WiFi signal from your network then outputs 3 wire pixel data.

4.1.1 Voltage in

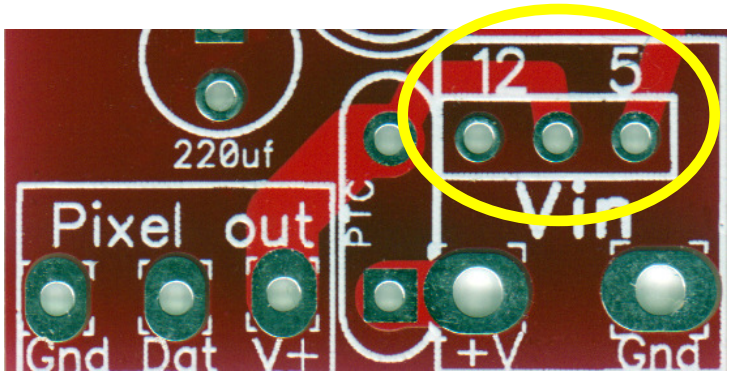
The voltage in is applied to the board on the Vin connections:



The positive voltage goes to the pad marked +V and the negative or ground goes to the "Gnd" pad. The voltage is usually protected with a polyswitch resettable fuse in the location marked PTC next to the Vin. We suggest a 30v 3A which will protect up to 3A through the board. Do not exceed this going through the board or the resettable fuse will "trip". You may need to consider feeding your pixel voltage externally to the pixels and not use the in to out on the WiFiPixel board.

4.1.1.1 Set the voltage in!

The Vin that you provide to the WiFiPixel is determined by the voltage requirement of the pixels that you use with it. The board passes the Vin through the polyfuse out to the "pixel out" so you must use the voltage that the pixel needs. Pixels are typically +5 or +12 and that must be set on the WiFiPixel board just above the Vin pads:



Place and solder a jumper wire (or install a 3 pin header and place a jumper on it) from the middle pin to either the left pin show for 12v in or from the middle pin to the right pin for 5v. Caution: if you set the jumper for 5V then put 12V into the board it will likely blow up the board. If you put the

jumper to 12V and put in 5V, the board will not function properly (the ESP will probably work but the pixels are unlikey to work properly).

Please MATCH the jumper setting to the voltage you use!!!

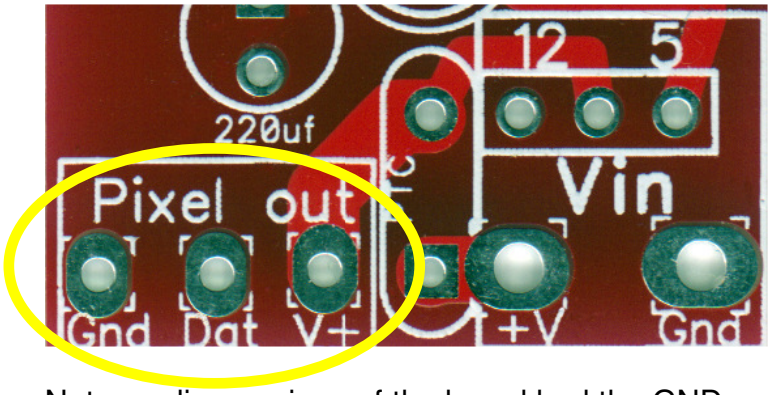
4.1.2 Data IN

The data sent to the WiFiPixel board over the WiFi connection is sACN E1.31 “DMX over network” data. More information can be found on the links provided later, or with a Google search.

4.1.3 Pixel connection OUT

The WiFiPixel board supports the most common “3-wire” pixels. These include the WS2811/2812 pixel types. The code in the ESP module is being updated all the time for different support so new 3-wire pixel formates can be supported in the future.

The output to the board is GND, Data, and V+ as shown:

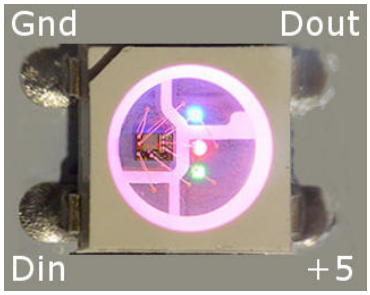


Note: earlier versions of the board had the GND and V+ the other way around. It might be a good idea to taek a close look at the board you get to see what order these pins are on the board you have.

4.1.4 Pixel signal examples

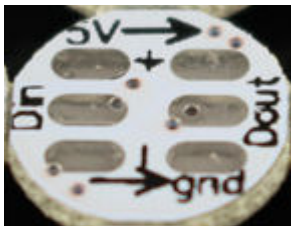
The following are SOME of the different pixels that can be driven by the WiFiPixel and how to determine the connections to the WiFiPixel. Keep in mind different pixels can have different wire colors, pin locations and different signal orders although for this type of pixel, they tend to have very consistent pinout as you can see comparing the above. You may need to refer to your specific pixel datasheet, or provider for details, or examine your pixels carefully for the clues as to how it is wired. Getting the wiring WRONG can be disastrous to the pixel. WS2811s are easily blown when wired wrong, but WS2812Bs are a bit more robust against being improperly wired.

4.1.4.1 Pixel



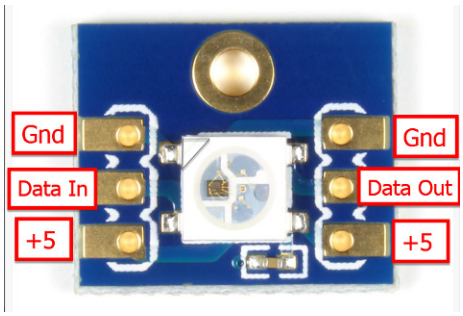
This is a WS2812B RGB pixel chip. Note how it has +5, ground, and data in (Din to connect to the WiFiPixel Dat output) as well as a data out (Dout) that provides the data stream to the next pixel in line. The WiFiPixel board is able to drive a chip like this but it is not very convenient to wire.

4.1.4.2 Pixel Board



Note how +5 is on the top, GND is on the bottom and DIN (that connects to the "Dat" pin on the WiFiPixel) is in the middle on the left side. The Dout (and voltages) connect to the next pixel (on the Din side).

4.1.4.3 Pixel "Breakout" board



The breakout board is similar to the pixel board above but provides a way to mount it and wire it. The "Dat" data out of the WiFiPixel board connects to the Data In connection of the breakout board, and the Data Out is the data stream to the Data In of the next pixel in line.

4.1.4.4 Pixel Strip

This is a pixel strip of RGB pixel chips (WS2812B in this case). Note the GND, DIN and +5V indicating where ground, data in, and +5V goes. Again the "Dat" output of the WiFiPixel board goes to the "DIN" on the strip. Note: This would be a very bad pixel to run on 12v!! Make sure the WiFiPixel is configured for 5V and you only apply +5 to it. Some strips run on +12v and these provide for a way to drop the strip 12v power to the 5v the chip needs.

4.1.4.5 12mm Pixel String

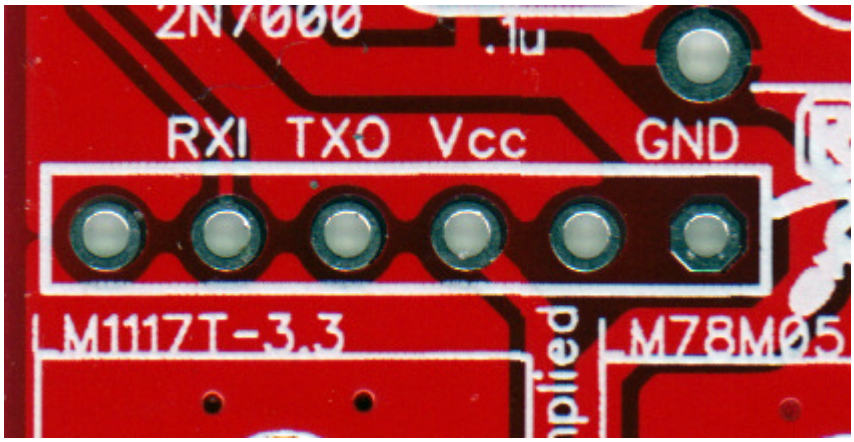
Looking inside this 12MM pixel, through the weatherproofing, you can barely see the connections marked 12V, DI and GND. The opposite side is the OUT side which will be GND, Dout, and 12V to the next pixel in line (if any). Obviously this type of pixel must be run on a WiFiPixel board configured for 12V with 12v power in. If one pixel has the information obscured, try looking at others until you find one that is readable.

4.1.4.6 NeoPixels

NeoPixels are simply WS2812 3-wire pixels on a circuit board. This can be driven by the WiFiPixel board as well as any other WS2812 pixel based device. They are clearly marked with data in/out and power connections.

4.1.5 Programming Port

The WiFiPixel provides a programming port for updating the code on the ESP module. It accepts standard serial TTL data same as that used to program some Arduino boards. The Arduino IDE is used to do the programming and the port is where you connect up the USB Serial TTL adapter output. Following is the programming port on the board:



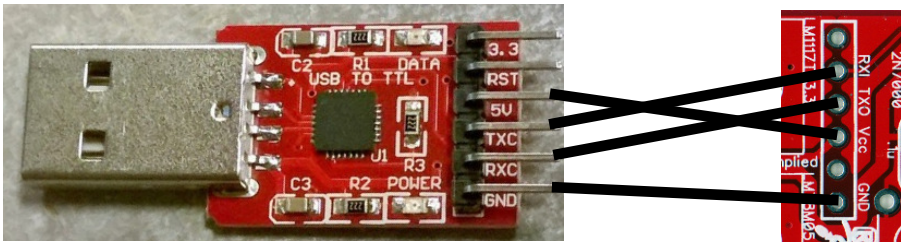
The Pinout is:

n/c	Not used on the WiFiPixel board
RX In	Serial TTL data in to the ESP from USB adapter
TX out	Serial TTL data out of the ESP to the USB adapter
Vcc	+5 from USB serial ttl adapter
n/c	Not used on the WiFiPixel board
Gnd	Ground from USB serial ttl adapter

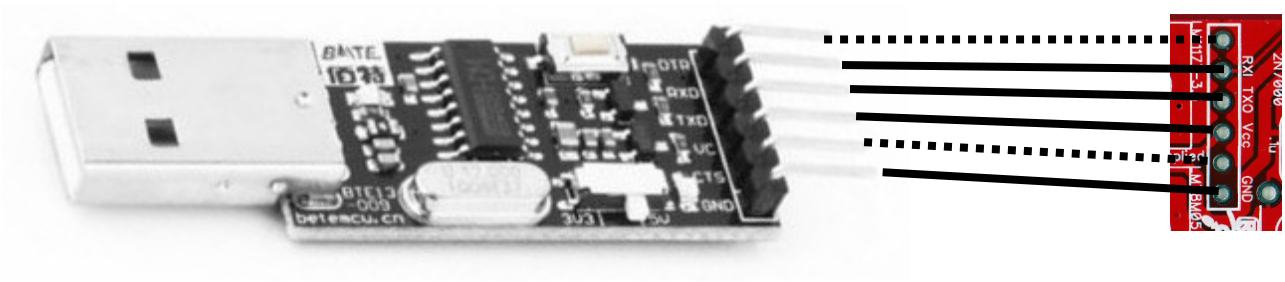
This port attaches to a USB to Serial TTL as indicated in the BOM. The following are some examples of how to connect with different USB serial adapters.

Note: some adapters reverse RX and TX so if you cannot communicate with the ESP module, try reversing the TX and RX wires.

4.1.5.1 DIGWDF CP2102 Adapter



4.1.5.2 CH340G



Note: the DTR out and CTS out from the CH340G may be connected to the corresponding n/c connections (indicated with dotted lines above) on the WiFiPixel Programming Port. This means a straight through 6 pin connector can be used. Remember that some USB adapters do reverse RX and TX which may require the pins to be swapped by the cable.

Details on using the programming port are discussed in the Programming the ESP8266 section that follows.

5. Programming the ESP8266

The ESP8266 module does not come with pixel capability programmed into it. You will need to set up the ESP module to operate with pixels. This is done using the Arduino programming environment (like the one used on Arduino Uno's, Pro Minis and the like) except with support added for the ESP8266. We mentioned this in the intro, but who reads that stuff so we will repeat the information here:

The ESP module is programmed using the Arduino IDE found here:
https://www.arduino.cc/en/Main/Software

with the ESP support added per Adafruit's guide here:
https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/using-arduino-ide

using the sketch from madsci1016 here:
https://github.com/madsci1016/ESPixelStick

or from forkineye here:
https://github.com/forkineye/ESPixelStick

Programming with a USB to TTL Serial adapter such as the CP2012 here:
http://digwdf.com/store/product.php?id_product=178

Additional information can be found on light animation sites like, www.diychristmas.org, etc. (search for "ESP8266" or "ESPixel").

The WiFiPixel board is completely software compatible with the ESP programming for the ESPixelStick as detailed on Shelby Merrick's site:
http://forkineye.com/ESPixelStick/

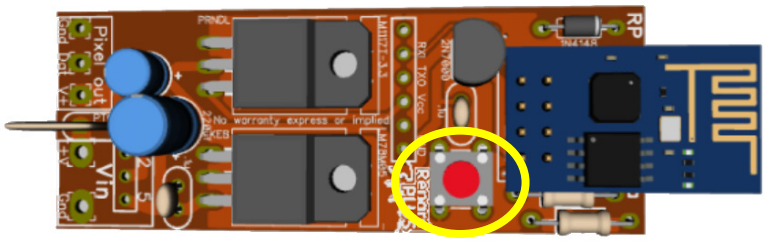
or Bill Porters site:
http://www.billporter.info/2016/01/18/espixel-gece-wifi-pixel-controller/.

Once the ESP has the pixel code in it, access to the pixels is handled through an on-board web server, and light animation data is sent using networking protocol sACN / E1.31 DMX. Your animation software must support E1.31 capability to work with the WiFiPixel. The Vixen animation software is a great example as it has such a capability either built-in or through an add-on.

5.1 WiFiPixel Programming Considerations

- It is a good idea to be sure you set the correct flash size of your ESP-01 module in the Arduino environment. The default is “512K (64K SPIFFS)”. The current ESP-01 modules mostly have 1M flash parts on them so if your modules do as well, you should set the flash size to the “1M (64K SPIFFS)” option (or larger if you have larger flash). Leaving the size at the default can result in the ESP module startup process crashing sporadically. This may be due to either code being programmed in the wrong place in the flash, or the ESP unintentionally accessing the possibly uninitialized flash regions. Once the setting matches the hardware, this possible issue is not seen.

- The WiFiPixel has a single button to put the ESP module into programming mode.



To enter programming mode, this button must be pushed BEFORE the WiFiPixel is powered on and stay pushed during the Arduino programming process in order to guarantee the programming process. Some modules have been seen to stay in programming mode if the button is released after power up and some will error out during the programming if the button is released.

- The USB Serial adapter is able to power the WiFiPixel in order to program the attached ESP module. The USB should not be used for the power source if pixels are attached to the WiFiPixel board as it could potentially burn up the USB adapter. It is best to leave pixels unconnected during programming unless you provide an external higher current power source that can power the board and pixels.

6. Final Test.

Now that you have your ESP programmed pixel wired up properly, the correct voltage configuration on the WiFiPixel, and the correct voltage in, it is time to try out the pixels.

You just need to set up the WiFiPixel as suggested in guides for the ESPPixel or ESPPIxelStick, set up your light animation software for E1.31 operation to the address you configure in the ESP, and set up a light run keeping in mind the following:

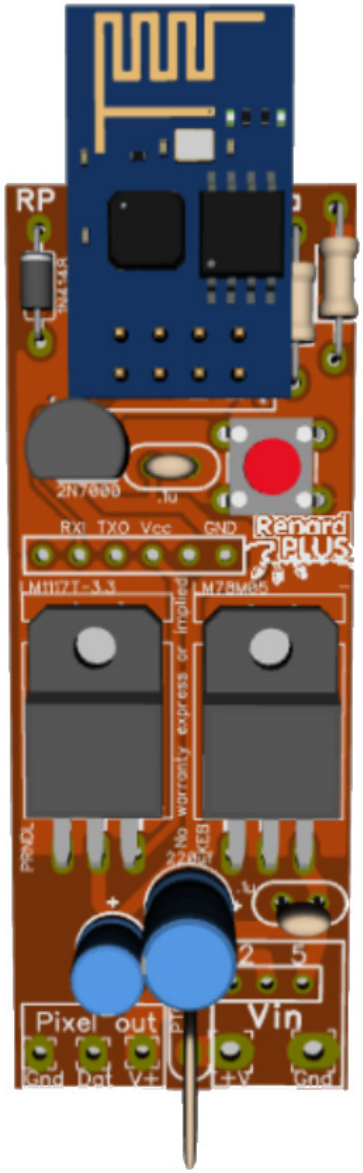
- With Light Animation Software that is not specifically designed for pixels, each pixel is usually THREE addresses (one for each Red, Green and Blue although not necessarily always in the order as it is pixel specific) so to be able to test all of your pixels in a string, you need to set up addresses for 3 x the number of pixels in your string, but you cannot exceed 512 addresses because E1.31 is DMX based and is limited to 512 addresses. The most you can do in a single string and address ALL colors on all pixels is 170 pixels (3x170 = 510). The good news is that you can continue your pixel string with another WiFiPixel on another e1.31 Universe and address as many universes as your software supports.

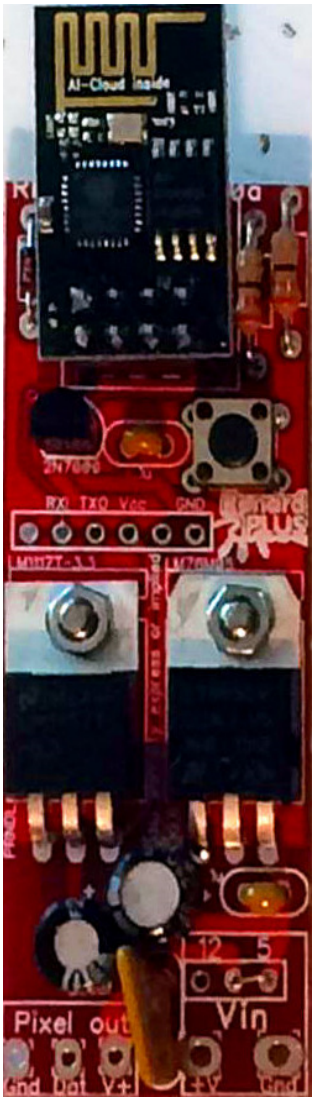
- Another thing to keep in mind is that with light animation software not specifically designed for pixels is that you need to turn on combinations of red, green and blue to get the other colors. White is equal Red, Green and Blue all on at usually equal levels to achieve white. Red + Blue will get you purple, and different levels of the combinations will get you the different shades of all of the colors your pixels are capable of displaying.

Have fun and try out different color ramps, and effects.

Congratulations, with a successful test, you have completed your build of your Renard Plus board and are ready for the wonderful world of light animation sequencing!

6.1 Picture of Finished Board (recommended configuration)





7. Notes

Use this page for YOUR notes about the boards.