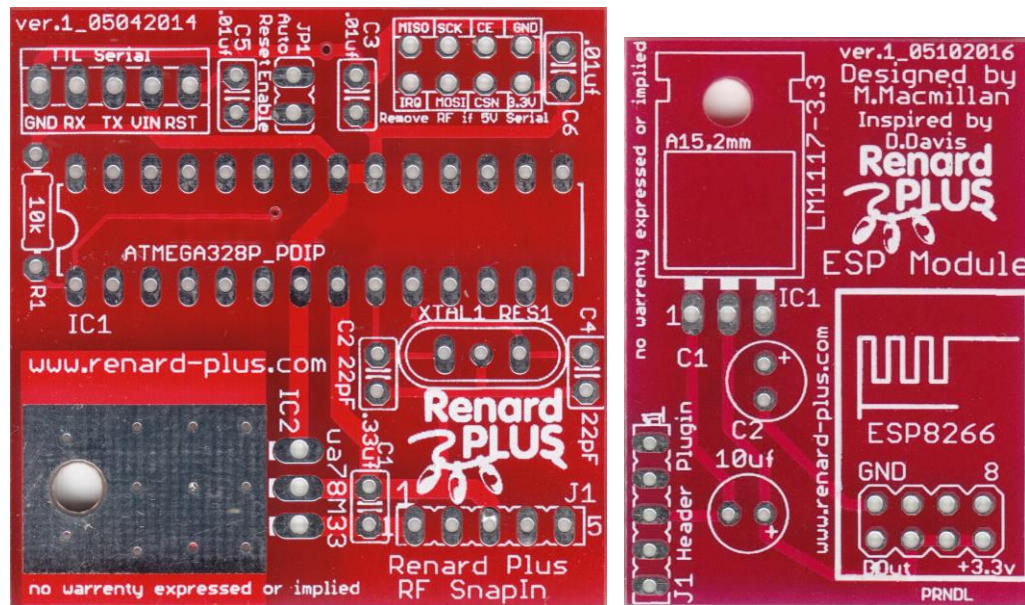




SnapIn Wireless Adapters

RF SnapIn / ESP SnapIn



Apr 2017

ESP Board Version 1.00 (v.1_05102016)
RF Board Version 1.00 (v. 1_05042014/ 1_09162014)
Document Rev 1.2

Renard-Plus, Salem, Oregon 97302
 © 2011-2017 Renard Plus. All rights reserved.
 Published 2017. Printed in United States

Renard-Plus ("Developer") has made every effort to ensure the accuracy of this document. Developer makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability and fitness for a particular purpose. The information in this document is subject to change without notice. Developer assumes no responsibility for any errors that may appear in this document.

The information contained herein is the exclusive and confidential property of Renard Plus, except as otherwise indicated. We wish to also thank the Do It Yourself Community for the inspiration it has given us in the development of this product.

Trademarks: the Renard Plus logo are trademarks of Renard Plus. All other trademarks acknowledged.

Table of Contents

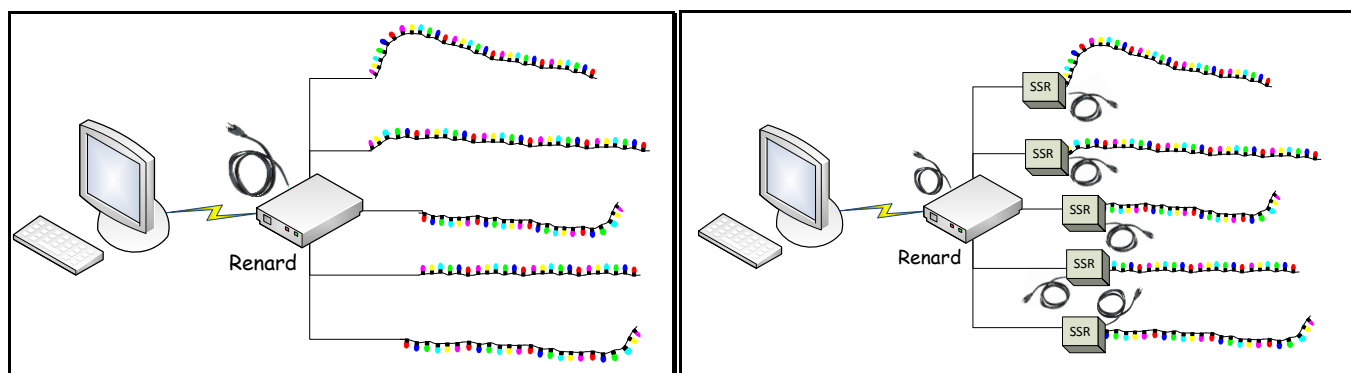
1. INTRODUCTION TO RENARD	3
2. OVERVIEW OF RENARD PLUS WIRELESS SNAPIN MODULES	4
2.1 ESP SNAPIN FEATURES.....	5
2.2 RF SNAPIN FEATURES	5
2.3 CONTROLLER POWER REQUIREMENTS FOR WIRELESS	6
3. ESP SNAPIN ASSEMBLY INSTRUCTIONS.....	7
3.1 ESP SNAPIN BOM / PARTS LIST	7
3.2 ESP SNAPIN PARTS ASSEMBLY.....	8
3.2.1 <i>First Things First - Inspection</i>	8
3.3 ESP SNAPIN BUILD	9
3.3.1 <i>Install Parts</i>	9
3.3.2 <i>ESP Snapin Initial Testing</i>	10
3.3.1 <i>ESP Snapin Install Wi-Fi Module</i>	11
3.3.2 <i>Picture of Finished ESP Board</i>	11
3.4 ESP SNAPIN FINAL STEPS AND TESTING	12
3.4.1 <i>ESP Snapin: Programming the ESP8266 module</i>	12
3.5 v.1_05102016 BOARD MOD FOR LATEST CODE.	12
4. RF (NRF) SNAPIN ASSEMBLY INSTRUCTIONS	15
4.1 RF SNAPIN BOM / PARTS LIST	15
4.1.1 <i>Selecting an NRF</i>	16
4.1.2 <i>NRF Transceiver Types</i>	17
4.1.3 <i>NRF Range Tests</i>	18
4.2 RF SNAPIN PARTS ASSEMBLY	18
4.2.1 <i>First Things First - Inspection</i>	18
4.3 RF SNAPIN BUILD.....	19
4.3.1 <i>Install Parts</i>	19
4.3.2 <i>RF Snapin Initial Testing</i>	21
4.3.3 <i>RF Snapin Install IC</i>	22
4.3.4 <i>Picture of Finished Board</i>	22
4.4 RF SNAPIN FINAL STEPS.....	23
4.5 RF SNAPIN: PROGRAMMING THE ATMEGA OVERVIEW.....	23
4.6 SELECTING A USB TO SERIAL TTL ADAPTER	23
4.6.1 <i>Adapter Requirements</i>	23
4.6.2 <i>Adapter Issues</i>	24
4.6.3 <i>Adapter Recommendations:</i>	24
4.7 INSTALL THE ARDUINO IDE.....	26
4.8 CONFIGURE THE SERIAL PORT	26
4.8.1 <i>Determine the COM Port #</i>	26
4.8.2 <i>Configure the IDE port</i>	26
4.9 KOMBY CODE FOR RF SNAPIN RECEIVER.....	27
4.10 PROGRAM	29
4.11 RF SNAPIN JUMPER SETTINGS / HEADERS.....	30
4.11.1 <i>Auto Reset Enable</i>	30
4.11.2 <i>RF SnapIn Header</i>	30
4.11.3 <i>NRF Module Connector</i>	30
4.11.4 <i>Serial TTL Arduino Programming Connector</i>	30
4.12 COMPUTER SETUP FOR RF SNAPIN.....	31
4.13 RF SNAPIN FINAL TESTING	31
5. NOTES	32

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](http://computerchristmas.com) '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 or a combination of these depending on the design.

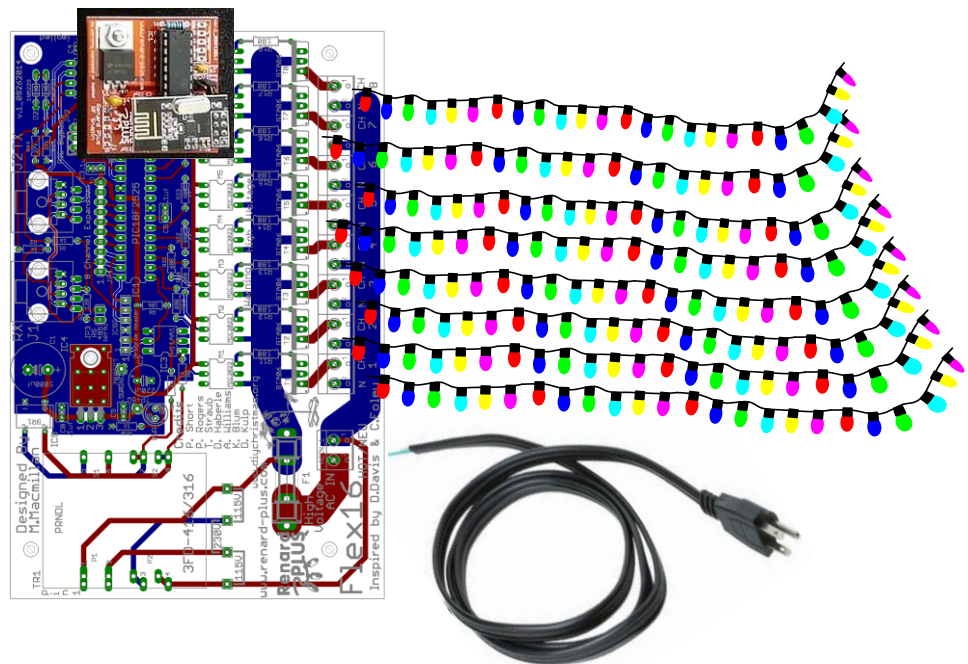
Renard is a DIY hobbyist effort and there is a vast amount of products and related peripherals to select from including the Renard Plus Strip Controller. To obtain a specific design, there might be “buy a parts kit and/or blank PCB” offering at a site (such as from www.renard-plus.com), “etch it yourself”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

2. Overview of Renard Plus Wireless SnapIn Modules

This guide covers the Renard Plus wireless SnapIn modules – RF Snapin and ESP Snapin. These boards are designed to convert a Renard Plus controller board (such as the TR8 Flex, RP16, RP32 and others in the Renard Plus controller line) to operate WIRELESSLY. A standard RP controller normally is interfaced to the sequencing system via a cable. A wireless Snapin module allows you to connect to the controller without the wire. Obviously there are still wires connecting the controller to power and to the lights it is controlling, but you can eliminate the data wire simplifying the hardware installation.

The wireless Snapin modules are available in two different flavors depending on how you want to implement the infrastructure. The RF Snapin is based on the NRF24L01+ RF module which provides direct control over the wireless operation with a corresponding NRF24L01+ transmitter at the computer. The ESP Snapin is based on the ESP8266 WiFi module which operates in a more traditional WiFi environment utilizing a standard wireless access point / router as the transmitter and implementing a mini web server in it for configuration. Both modules operate in the 2.4GHz band and either plugs into the Snapin Module connector on Renard Plus controllers. NOTE: ESP and NRF do NOT communicate with each other- the wireless protocols are completely different.

Example:



2.1 ESP Snapin Features

Feature		Detail
Name		Renard Plus ESP SnapIn
Target use		Wireless Receiver for Renard Plus controllers based on ESP8266 ESP-01 WiFi modules
Channel Count		n/a
Power input		From Renard Plus
Power output		n/a
Dimmable?		n/a
Status Indicators?		No
Channel Indicators?		n/a
Control Input – Renard		YES
Control Input – DMX		YES
Daisy-chain output		Wired daisy chain from Renard Plus is supported
Wireless		YES
On board programming		No but configuration is done over wireless
Enclosure		n/a
Heatsink?		No

2.2 RF Snapin Features

Feature		Detail
Name		Renard Plus RF SnapIn
Target use		Wireless Receiver for Renard Plus controllers based on NRF24L01+ modules
Channel Count		n/a
Power input		From Renard Plus
Power output		n/a
Dimmable?		n/a
Status Indicators?		No
Channel Indicators?		n/a
Control Input – Renard		YES
Control Input – DMX		YES
Daisy-chain output		Wired daisy chain from Renard Plus is supported
Wireless		YES
On board programming		Yes through TTL Serial
Enclosure		n/a
Heatsink?		No

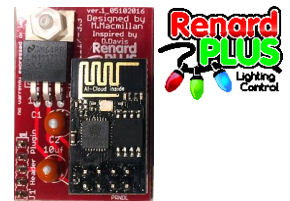
2.3 Controller Power Requirements for Wireless

Most of the Renard Plus controllers have very modest power requirements typically under 200-400ma for operation. This allows for the use of lower power (i.e. less expensive) transformers on controllers with onboard power. Adding wireless can DOUBLE the current requirements of the board. The ESP module itself can use upwards of 300+ma of 3.3v at startup and peaks when running full speed. The nature of the streaming data required for light control means the wireless will be running at full power as long as show data is being handled.

If you plan to use wireless, you need to be sure the transformer you use on the board is capable of the extra current that streaming wireless requires when utilizing the wireless SnapIn options.

We have seen that it is not uncommon to use the lower capacity transformers when building controllers. Those simply do not provide enough current to run the controller with wireless. Our recommendation is to use the higher capacity transformer option of the **838-3FD-416** (8v, 800ma) that controller build guides specify instead of the lower capacity 838-3FD-312 (6.3v 400ma) or 838-3FD-316 (8v 300ma) options that are available.

If you are using lower capacity transformers, you will need to change to a higher power/current transformer to get reliable wireless operation.



3. ESP Snapin Assembly Instructions

This chapter covers the construction of the Renard Plus ESP SnapIn module.






It approaches the task 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 board 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 ESP Snapin BOM / Parts List

The following is the Bill Of Material for building the Renard Plus ESP SnapIn.

Note: If you did not obtain a kit that some lighting vendors may offer, Mouser is the most convenient place to order most of 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 for alternatives. Some parts, like the ESP module are generally only available from Ebay or Aliexpress.

Picture	Designators	Description	Qty	Mouser P/N
	C1, C2	10uf 16v radial lead tantalum cap	2	581-TAP106K016GSB http://www.taydaelectronics.com/capacitors/tantalum-capacitors/10uf-16v-radial-tantalum-capacitor.html
	IC1	LM1117-3.3 3.3v LDO regulator TO-220	1	511-LD1117V33 http://www.taydaelectronics.com/lm1117t-lm1117-low-dropout-voltage-regulator-ic-3-3v.html
	J1	5 pin female 1x5 connector (5Fx1) 2.54mm spacing	1	517-929870-01-05-RA http://www.taydaelectronics.com/connectors-sockets/pin-headers/5-pin-2-54-mm-single-row-female-pin-header.html
	ESP Module	8 pin female 2x4 connector (4Fx2)	1 ³	992-4FX2-254MM (see note later in this section for low cost Tayda Electronics option)
	ESP Module	ESP8266 ESP-01 model programmed for	1	EBay or Aliexpress



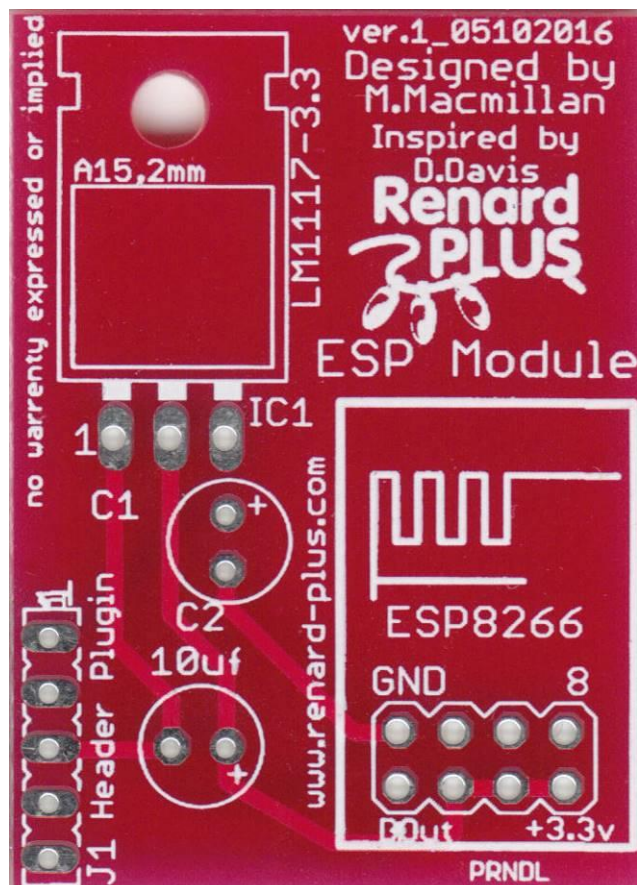


3.2 ESP Snapin Parts Assembly

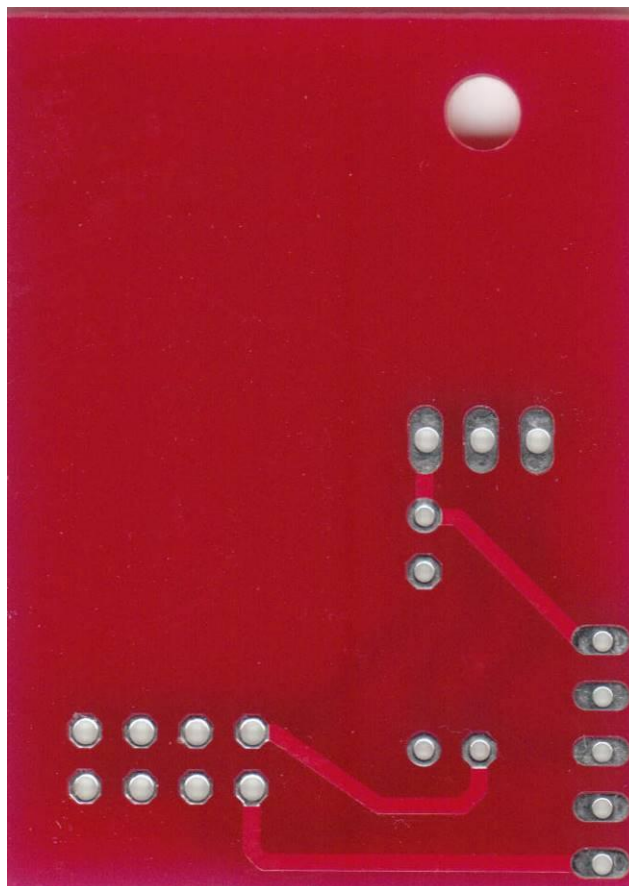
The Renard Plus wireless SnapIn boards are fairly simple devices to assemble and test. It is easiest if you follow these instructions, checking off steps as they are performed. This will lead you through the assembly installing components from shortest/smallest to tallest.

3.2.1 First Things First - Inspection

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



ESP Front



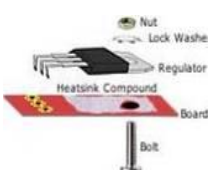
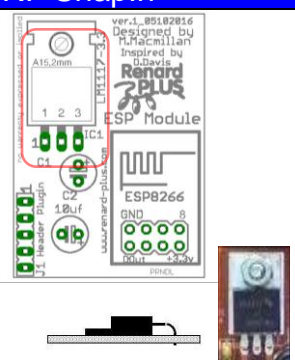
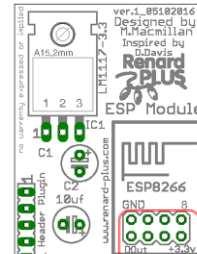
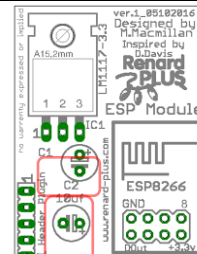
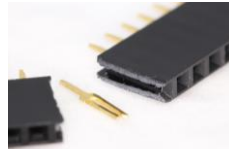
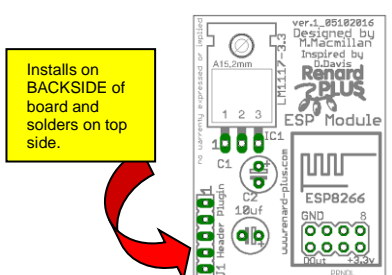
ESP Back

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 appropriate assembly guide as follows in the next sections.

3.3 ESP SnapIn Build

3.3.1 Install Parts

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. See the Tools and Parts ID document on www.renard-plus.com for details.

Step	Instructions	RF SnapIn
1 <input type="checkbox"/>	<p>Install the 3.3v linear regulator LM1117 3.3v at location IC1 forming the leads as indicated below. 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 and after inserting the leads into the proper holes, secure the IC with a 4-40 bolt, #4 lock-washer, and 4-40 nut. Solder and clip leads.</p>  <p><i>Note: if you cannot find short 4-40 bolts, you can use a longer one and clip it with diagonal clippers after bolting it down.</i></p>	
2 <input type="checkbox"/>	<p>Install 4Fx2 connector on the top side of the board for the ESP Module. Solder</p>	
3 <input type="checkbox"/>	<p>Install the two caps at C1 and C2. Solder and clip leads.</p> <p><i>Note: these caps have a specific orientation. Look for a + mark (and the longer lead) which indicates the lead that goes in the pad marked with a + on the board silkscreen.</i></p>	
4 <input type="checkbox"/>	<p>Install 5Fx1 connector on the BOTTOM side of the board at J1. Solder</p> <p><i>Note: you may need to cut down a larger connector to 5 pins. Using diagonal cutters, cut on the 6th pin position to get 5 pins.</i></p> 	



3.3.2 ESP Snapin Initial Testing

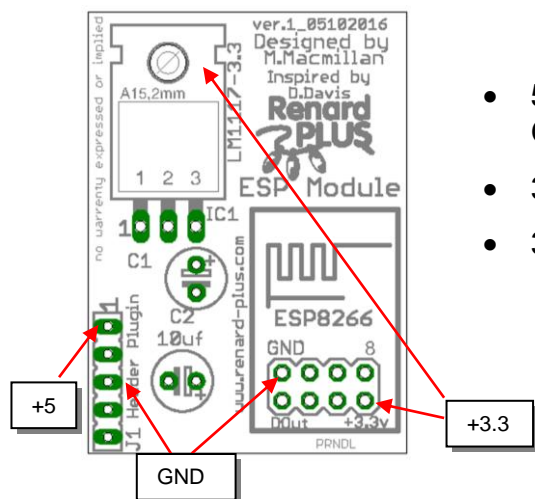
At this point you have completed the assembly of the board (less the ESP module) and you should gently clean the board of any residue and inspect for solder bridges or cold solder joints. What you are looking for are any solder bridges especially around the IC's and other closely spaced parts, or pins that are not fully and cleanly soldered.

If you have the wireless module installed – **remove now**.

Plug the board into the RenW Snap In connector on a working Renard board. You will need to remove the jumper that is likely on the header there. Power up the controller.



Verify the power LED on the RP Controller lights up. If you see lights, use your volt meter (e.g. DMM) and on the RF SnapIn board to verify the following:




- 5 volts on J1 pin 1 (top most pin) and pin 3 the GND pin (center pin).
- 3.3v on the regulator tab to GND.
- 3.3v on the ESP header 3.3v pin

If the voltage does NOT measure around +5 (4.8 – 5.2) or 3.3v (3.2 – 3.4) at the test points, remove power and start troubleshooting. Make sure the power supply on the Renard Plus controller is correct. Look for solder bridges around the regulator. Double check the regulator number to make sure it is what you expect (something like 78M33). Look for cold solder joints – retouching all solder connections will often help solve issues like this.

When power measures properly, disconnect power and continue assembling.

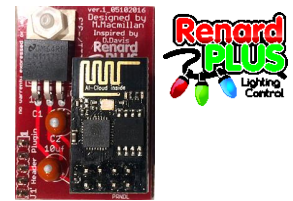
3.3.1 ESP Snapin Install Wi-Fi Module

Note: Before handling the ESP8266 module, touch the bottom of the board or use a conductive wrist-strap attached to the board.

Step	Instructions	RF SnapIn
5 <input type="checkbox"/>	<p>Install the ESP8266 wireless module on the ESP Module connector.</p> <p><i>Note: the module does have a specific orientation and should be installed as the board silkscreen shows.</i></p>	

3.3.2 Picture of Finished ESP Board





3.4 ESP Snapin Final Steps and Testing

At this point you will have now completed the installation of all of the parts to the controller. Again, it is a good idea to gently clean off any final soldering residue and then visually inspect the board and check to make sure there are no solder bridges between the solder pads, and that the solder joints are all a good quality.

3.4.1 ESP Snapin: Programming the ESP8266 module

With the board built, you need to get the ESP8266 module programmed for blinky operation. The process is summarized as:

- Compile/flash the module with Shelby (aka user “forkineye” on diychristmas.org) or Bill Porter’s version of the ESP firmware with the Arduino IDE. *Note: See following board mod required for the v.1_05102016 version of board to use the latest revision of Shelby’s code.*
- Power up the module and allow it to attach on to the wireless network you will use for blinky
- Log into the module via the IP address using a browser on a system on that wireless network (either wireless or wired!)
- Set and save the configuration options (details are in the docs linked below)
- Plug the ESP into the ESP Snapin board
- Plug the ESP Snapin into the controller
- Test the controller for proper function (note: communications are handled using E1.31 sACN protocol that you need to set in your sequencing software like Vixen +

The best procedure for handling this is detailed on Dave’s (aka Dirk Nerkle) digwdf.com store site where we recommend using a “Flashmater” ESP programming board:

http://digwdf.com/store/product.php?id_product=210

and USB to serial TTL adapter CP2102:

http://digwdf.com/store/product.php?id_product=178.

The instructions for programming the ESP module is found within these guides:

http://digwdf.com/store/attachment.php?id_attachment=41

(look for ESP module programming instructions)

http://digwdf.com/store/attachment.php?id_attachment=37

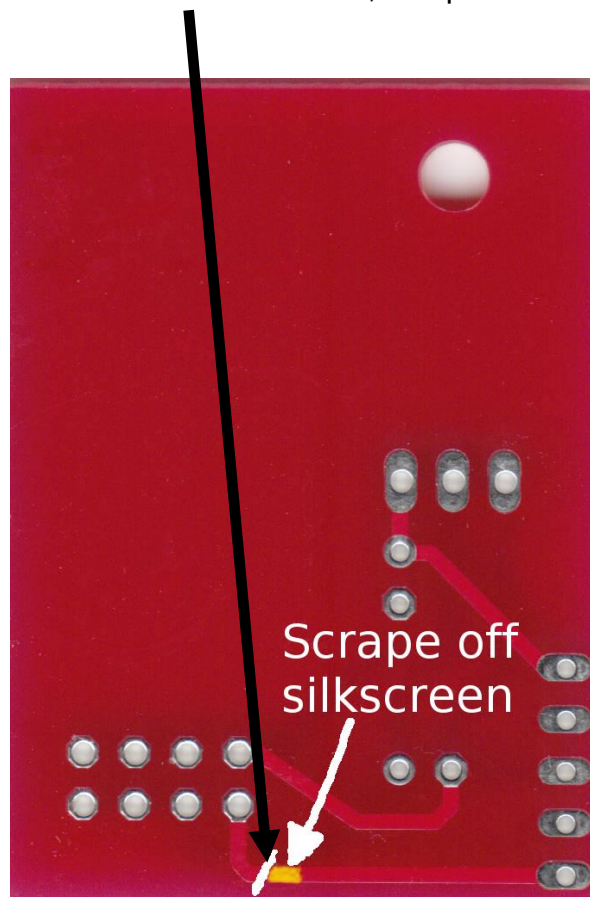
(contains additional information on using the ESP module and configuring it)

More information may also be found on the diychristmas.org website.

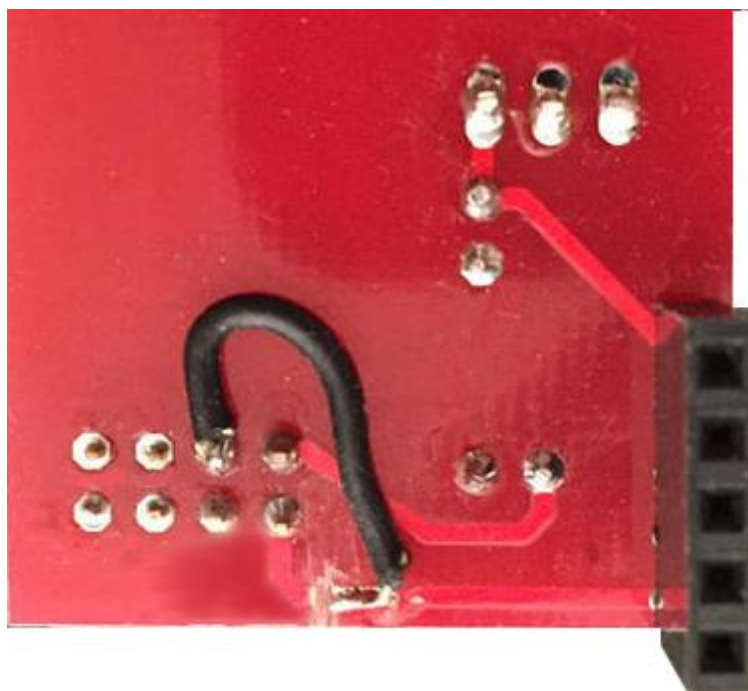
3.5 v.1_05102016 Board Mod for latest code.

If you are using the v.1_05102016 of the board, and will be using the latest (mid 2016 and later) revision of the Shelby code, you will need to do a small wire mod. The Bill Porter uses and the older Shelby code used the serial out of the ESP8266 to pass the data. The newer Shelby code now uses a different GPIO pin for the data stream output. To use this code with the v.1_05102016 board, you need to cut a trace and add a wire to route the new signal.

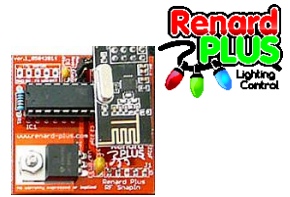
First on the back side of the board cut this trace. Next, scrape off the silkscreen from the trace.



Then attach a wire from the scrapped trace to the pin of ESP8266 connector as follows:



<End of ESP Snapin Module Instructions>



4. RF (NRF) SnapIn Assembly Instructions












This chapter covers the construction of the Renard Plus NRF SnapIn module. It approaches the task 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 board when you are done. Additional information and guides on techniques and tools can be found in the “Tools and Parts ID Guide” at:

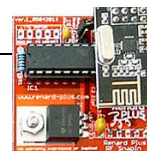
www.renard-plus.com/files/Tools_and_Parts_ID_Guide.pdf





4.1 RF SnapIn BOM / Parts List

The following is the Bill Of Material for building the Renard Plus RF SnapIn. The link to the Mouser project is: <http://www.mouser.com/ProjectManager/ProjectDetail.aspx?AccessID=xxxxx>

Note: If you did not obtain a kit that some lighting vendors may offer, Mouser is the most convenient place to order most of 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 for alternatives.

Picture	Designators	Description	Qty	Mouser P/N
	R1	10k ohm resistor 1/8 or 1/4 watt	1	299-10k-RC http://www.taydaelectronics.com/resistors/1-4w-metal-film-resistors/10-x-resistor-10k-ohm-1-4w-1-metal-film-pkg-of-10.html
	C1	.33uf / 330nf ceramic cap	1	81-RCER71H334K1A2H3B http://www.taydaelectronics.com/capacitors/monolithic-ceramic-capacitor/0-33uf-50v-multilayer-ceramic-capacitor.html
	C2, C4	22pf ceramic cap	2 ¹	81-RCE5C1H220J0A2H3B http://www.taydaelectronics.com/capacitors/monolithic-ceramic-capacitor/22pf-50v-multilayer-monolithic-ceramic-capacitor.html
	C3, C5, C6	.1uf / 100nf ceramic cap	3	81-RDER71H104K0S1C03 http://www.taydaelectronics.com/capacitors/monolithic-ceramic-capacitor/0-1uf-50v-multilayer-ceramic-capacitor.html
	IC1	28 pin IC Socket (optional)	1 ²	571-1-390261-9 http://www.taydaelectronics.com/connectors-sockets/sockets/dip-sockets/28-pin-dip-ic-socket-adaptor-solder-type.html
	J1	5 pin female 1x5 connector (5Fx1) 2.54mm spacing	1 ³	517-929870-01-05-RA http://www.taydaelectronics.com/connectors-sockets/pin-headers/5-pin-2-54-mm-single-row-female-pin-header.html
	NRF Connector	8 pin female 2x4 connector (4Fx2)	1 ³	992-4FX2-254MM (see note for low cost Tayda Electronics option)
	(Opt) TTL Serial	5 pin male or female 1x5 (5Fx1 is recommended)	1 ^{2 & 3}	517-929870-01-05-RA http://www.taydaelectronics.com/connectors-sockets/pin-headers/5-pin-2-54-mm-single-row-female-pin-header.html
	JP1	2 pin male 1x2 header (2Mx1)	1	571-826926-2 http://www.taydaelectronics.com/connectors-sockets/pin-headers/40-pin-2-54-mm-single-row-pin-header-strip.html
	Auto Reset	Shunts / Jumper Plug	1	649-63429-202LF http://www.taydaelectronics.com/connectors-sockets/pin-headers/mini-jumper-2-54mm-gold-plated-closed-cover.html
	IC1	Atmega 328P-PU microcontroller with Arduino bootloader	14	556-ATMEGA328P-PU http://www.taydaelectronics.com/atmega328p-pu-atmega328-microcontroller-ic.html



	IC2	TO-220 3.3v regulator	15	595-UA78M33CKCS (see note for details on a possible Tayda substitution with tweak)
	XTAL1/RES1	2 pin Crystal -16.00MHz	16	774-ATS16A http://www.taydaelectronics.com/crystals-resonators-oscillators/crystals/16-000-mhz-16-mhz-crystal-hc-49-s-low-profile.html
	XTAL1/RES1	3 pin Ceramic Resonator - 16.00MHz	16	81-CSTLS16M0X53-B0 http://www.taydaelectronics.com/crystals-resonators-oscillators/ceramic-resonators/ceramic-resonator-16-0mhz-3-pins.html
	NRF	NRF24L01+ Wireless module	1	See section on Selecting an NRF

¹ Required for 2 pin crystal use, but omit if ceramic resonator used. See Note 6.

² Optional but usually recommended

³ A single one of these: <http://www.taydaelectronics.com/connectors-sockets/pin-headers/40-pin-2-54-mm-single-row-female-pin-header.html> can be used to make these three connectors. For the 2x4, just cut two 1x4 and solder them in side-by-side.

⁴ Must be pre-programmed with an Arduino Bootloader. Most 328P-PU parts are not pre-programmed including the Mouser and Tayda parts! Amazon, Adafruit, Ebay (e.g. <http://www.ebay.com/itm/161401449351>) and others offer parts with the bootloader burned in, or Google "Bootload Atmega328p" for instructions if you want to go it yourself.

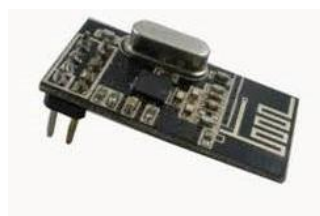
⁵ Pin out is "Vin GND 3.3vOUT". Tayda offers an LM1117T-33 LDO: <http://www.taydaelectronics.com/lm1117t-lm1117-low-dropout-voltage-regulator-ic-3-3v.html> but the pinout is "GND 3.3vOUT Vin" so it will not drop directly in. To use this part, you would need to do a little work to the regulator. See: <http://www.diychristmas.org/vb1/showthread.php?330-How-to-Use-an-LDO-Voltage-Regulator-in-a-standard-voltage-regulator-board> for details on using the LDO part. Hint:

⁶ Choose either a crystal + 2x22pf caps, or 1 resonator.



4.1.1 Selecting an NRF

The NRF24L01+ module is a wireless transceiver (transmit and receive) available in a wide range of styles and capabilities. Depending on your layout and controller positions, you can use any combination of modules you see fit. Range and reliability will change depending on a set of factors that cannot be known for your particular







situation. If you follow the recommendations, you should have a successful setup.



4.1.2 NRF Transceiver Types

The various modules have various sources and varying prices. The following are the variations available:

Name	Picture	Where	Notes
green “?”		http://www.ebay.com/itm/2PCS-2-4GHz-NRF24L01-Wireless-Transceiver-Module-NEW-/110929270444	10 pins. Shortest range NOT RECOMMENDED – Will NOT work.
Black “zig-zag”		http://www.komby.com/nrf24l01-24-ghz-rf http://www.ebay.com/itm/10pcs-NRF24L01-2-4GHz-Antenna-Wireless-Transceiver-Module-for-Arduino-New/400594940658? www.ebay.com/itm/330808881863	8 pins. Best range for inexpensive RX, great price, can be sensitive to power ripple.
Green PA		http://www.ebay.com/itm/nRF24L01-PA-LNA-wireless-communication-modules-w-antenna-2-4GHz-2Mbps-/110980385971 http://dx.com/p/nrf24l01-pa-lna-wireless-communication-modules-w-antenna-for-arduino-148822	8 Pins. Higher cost than the non-PA version because it is amplified for TX and uses an antenna. These will provide longer range but are 5x to 10x the price of the “zig-zag” modules. You should use one of the PA modules for the TX. Using these for RX increases the receive range some (because of the better antenna) but the TX amplifier reduces receive sensitivity so this is not best for RX. This one may need the two extra pins on the board manually connected to ground to operate.
Black PA		http://www.komby.com/nrf24l01-24-ghz-rf-pa-with-antenna http://www.ebay.com/itm/151143530299 NRF24L01+PA+LNA SMA Antenna Wireless Transceiver communication module	8 pins. Note: You may need to provide an antenna depending on where you get this. The two listed here both include the antenna.

4.1.3 NRF Range Tests

These are results obtained testing with the RFCOLOR2_4 board. Your results will vary, but should give a general idea of what works and what does not.

TX	RX	Range
Green “?”	Green “?”	30 feet
Black “zigzag”	Black “zigzag”	200 feet
Black PA	Green “?”	400 feet
Black PA	Black “zigzag”	600 feet
Black PA	Green PA (inside car)	>1000 feet
Green PA	Black PA (inside car)	>1000 feet

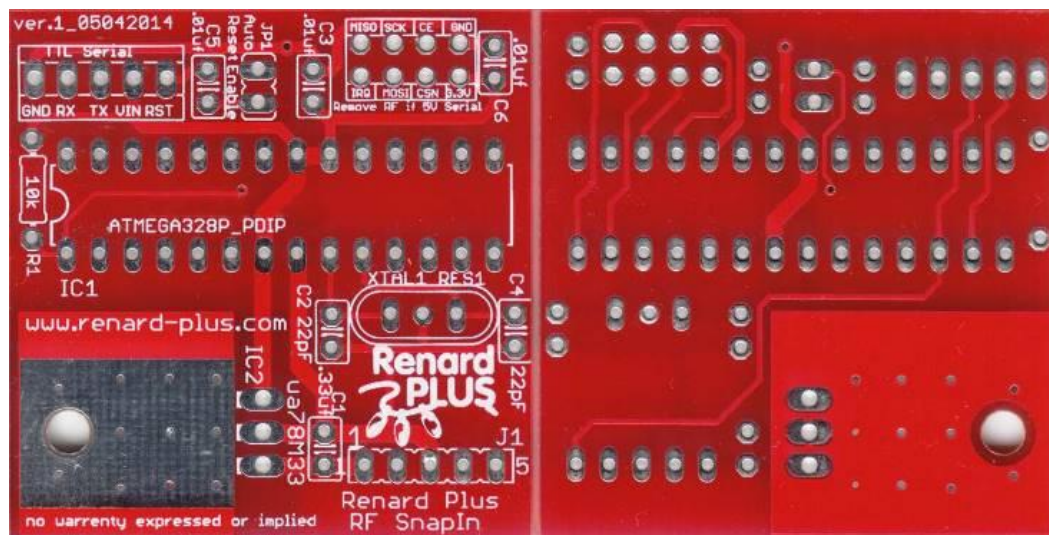
Here is the recommendation: in general use the Black “zigzag” modules for receivers and either of the “black PA” or “green PA” modules for the transmitter. The Green “?” modules are bad at transmitting and only OK at receiving. For longer range, use a “PA” module in both TX and RX.

4.2 RF Snapin Parts Assembly

The Renard Plus wireless SnapIn boards are fairly simple devices 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.

4.2.1 First Things First - Inspection

- 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.



RF Front

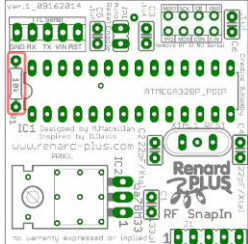
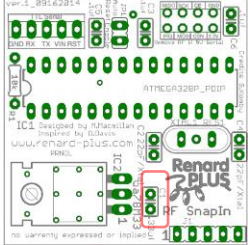
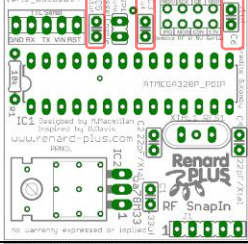
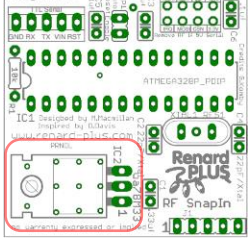
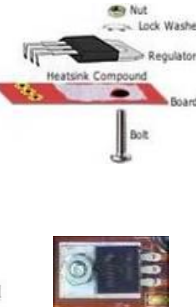
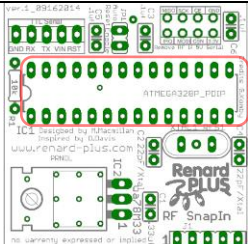

RF Back

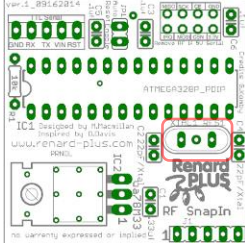

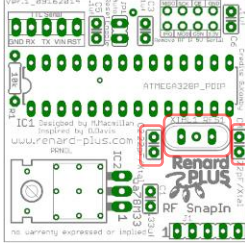
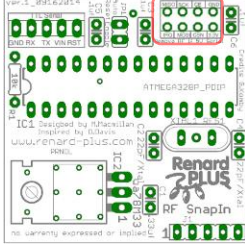
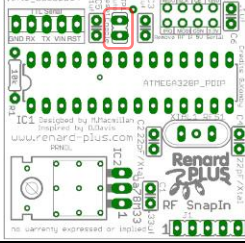
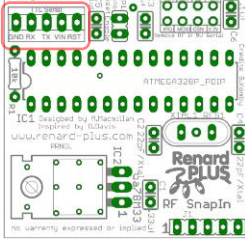
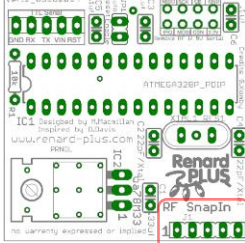

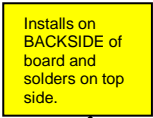
- Next inspect and sort out the various parts for the board. Make sure you understand which parts are which, and things like resistor codes and component orientation. A separate document on these concepts is available at www.renard-plus.com/files/Tools_and_Parts_ID_Guide.pdf and on other resource sites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

4.3 RF SnapIn Build

4.3.1 Install Parts

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. Other parts may or may not have a specific orientation. The instructions column will have a note to indicate any parts that have specific handling or not. If you do not see a note, assume that the part (like a resistor) does not need special handling. See the Tools and Parts ID document on www.renard-plus.com for details.

Step	Instructions	RF SnapIn
1 <input type="checkbox"/>	Install 10k ohm resistor (brown-black-orange) at locations R1. Solder and clip leads.	
2 <input type="checkbox"/>	Install one .33uf cap at location C1. Solder and clip leads. <i>Note: Bypass caps do NOT have a specific orientation. .33uf is also known as 330nf.</i>	
3 <input type="checkbox"/>	Install the three .1uf (usually marked 104) capacitors at locations C3, C5, C6. Solder and clip leads. <i>Note: Bypass caps do NOT have a specific orientation. .1uf is also known as 100nf.</i>	
4 <input type="checkbox"/>	Install the 3.3v linear regulator 78M33 at location IC2 forming the leads as indicated below. 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 part and after inserting the leads into the proper holes, secure the IC with a 4-40 bolt, #4 lock-washer, and 4-40 nut. Solder and clip leads. <i>Note: if your 4-40 bolts are long, you can use a longer one and clip it with diagonal clippers after bolting it down.</i>	 
5 <input type="checkbox"/> OPT	Optional: Install the 28 pin IC socket at location IC1. Solder. <i>Note: Recommended</i> Pin 1 of the IC socket is on the end, closest to the notch.	 

Step	Instructions	RF SnapIn	
6a <input type="checkbox"/>	Option a: install a three pin 16mhz cermamic resonator at XTAL. Solder. Leave C2 and C4 empty.		
-or-	<i>Note: Resonators do not have a specific polarity and may be installed in either direction.</i>		
6b <input type="checkbox"/>	Option b: install two pin 16 mhz crystal at XTAL and two 22pf capacitors at C2 and C4. Solder and clip leads.		
7 <input type="checkbox"/>	Install 8 pin (2x4) FEMALE header on the top side of the board for the NRF module. Solder.		
8 <input type="checkbox"/>	Install 2 pin (1x2) male header at JP1. Solder.		
9 <input type="checkbox"/> OPT	Optional: Install 5 pin (1x5) male or female header at location "TTL Serial". Solder. <i>Note: some leave this connector off, and use a male connector on the USB Serial TTL cable to temporarily connect to program the receiver sketch.</i>		
10 <input type="checkbox"/>	Install 5 pin (1x5) FEMALE header at location J1 on the BACK side of the board. Solder on top side. <i>Note: you may need to cut down a larger connector to 5 pins. Using diagonal cutters, cut on the 6th pin position to get 5 pins.</i>		 

4.3.2 RF Snapin Initial Testing

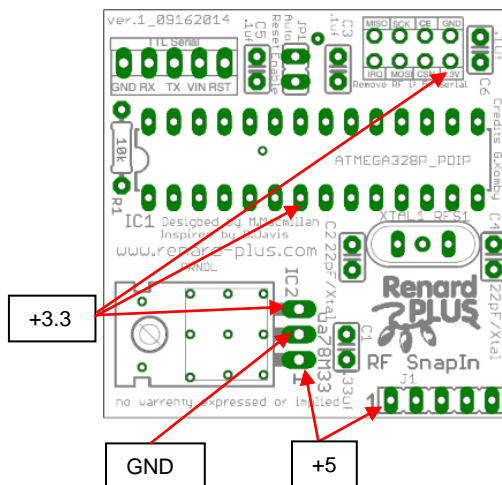
At this point you have completed the assembly of the board (less ICs and modules) and you should gently clean the board of any residue and inspect for solder bridges or cold solder joints. What you are looking for are any solder bridges especially around the IC's and other closely spaced parts, or pins that are not fully and cleanly soldered.

If you have the IC or wireless module installed – **remove now**.

Plug the board into the RenW Snap In connector on a working Renard board. You will need to remove the jumper that is likely on the header there. Power up the controller.

Warning
High Voltages Present

Verify the power LED on the RP Controller lights up. If you see lights, use your volt meter (e.g. DMM) and on the RF SnapIn board to verify the following:



- 5 volts DC between the 5V in to the regulator on pin 1 (bottom most pin) and pin 2 the GND pin (center pin).
- Regulator out pin 3 (top most pin) to GND pin 2 (middle pin) shows 3.3v.
- Pin 2 on the NRF header is reading 3.3v
- Pin 7 on the 28-pin socket (Atmega) reads 3.3v

If the voltage does NOT measure around +5 (4.8 – 5.2) or 3.3v (3.2 – 3.4) at the test points, remove power and start troubleshooting. Make sure the power supply on the Renard Plus controller is correct. Look for solder bridges around the regulator. Double check the regulator number to make sure it is what you expect (something like 78M33). Look for cold solder joints – retouching all solder connections will often help solve issues like this.

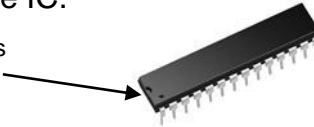
When power measures properly, disconnect power and continue assembling.

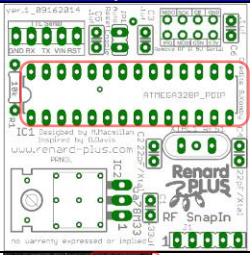

4.3.3 RF Snapin Install IC

Note: Before handling any IC, touch the bottom of the board or use a conductive wrist-strap attached to the board.

IC's pins are numbered from 1 to the number of pins counter clockwise with pin 1 being just to the right of either a notch or dimple on the IC.

Pin 1 of the IC socket is on the end, closest to the notch.



Step	Instructions	RF Snapin
11 <input type="checkbox"/>	Install the 28 pin Atmega328P-PU microprocessor (with Arduino Bootloader pre-burned) at IC1. <i>Note: Pin one goes toward the pin one / notch on the silkscreen. That is toward the right edge of the board as shown.</i>	
12 <input type="checkbox"/>	Install the NFR24L01+ wireless module in the NRF connector	

4.3.4 Picture of Finished Board

With and without NRF24L01+ wireless module.



Note: Pictured with the Ceramic Resonator used so caps to right and left of it are NOT populated.



Note: Pictured with the NRF24L01+ module populated.



4.4 RF Snapin Final Steps

At this point you will have now completed the installation of all of the parts to the controller. Again, it is a good idea to gently clean off any final soldering residue and then visually inspect the board and check to make sure there are no solder bridges between the solder pads, and that the solder joints are all a good quality.

4.5 RF Snapin: Programming the Atmega OVERVIEW

Your board is built, the power is tested, the Atmega is onboard and you are now ready to program the receiver sketch. The process to get the sketch on the board is very simple. The overview of the process follows and detailed in following sections:

- Use an “appropriate” USB to serial TTL adapter and drivers
- Install and run the Arduino IDE
- Configure the serial port of your device in the IDE. Note: varies from device to device
- Load the sketch (download the Komby RFSHOWControl code) and set appropriate options
- Program

4.6 Selecting a USB to serial TTL adapter

This step is the most bewildering and problem fraught part of the whole Arduino scheme. There are so many options available on the market with so many different capabilities, different pin-outs, and so many unscrupulous manufacturers or inaccurate descriptions that you can end up with something that won't work or requires special handling to get around the difficulties.

4.6.1 Adapter Requirements

What you need is a 5 or 6 pin USB to serial adapter that works with the O/S you plan to use to run the Arduino IDE, with the right outputs for an Arduino interface. You must have outputs that include:

Power (3.3v is best for all Arduino situations due to the NRF 3.3V requirements, but designs that include a 3.3v regulator for the NRF will work from 5V. If you are unsure, just be sure to unplug the NRF while updating the sketch.), GND, TX, RX, and CTS / RTS or DTR (data terminal ready)

Note: CTS is not a requirement but is common on some designs along with DTR. This signal is needed to reset the Atmega to initiate the sketch programming via the bootloader.) RST is also common but it has been found to be a reset IN to the USB board itself, not the needed reset to do auto-programming via the Arduino IDE!

The pin-out of the RF SnapIn Serial TTL input is as follows:

1	2	3	4	5
GND	RX	TX	VIN	RST

If you examine the descriptions of an adapter, it is best try to try to get as close to this pin out as possible or you will need to make an interconnect cable (or rewire the included one) to allow the board to plug in. RTS/DTR or CTS is the key to being able to auto-program but it will NOT be labeled “RST” on the USB adapter as that signal is actually used to reset the USB converter chip (like a CP2102) and is NOT an output as we need.



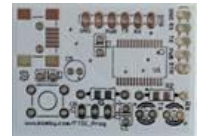
4.6.2 Adapter Issues

Following is a list of issues we have experienced with various boards:

1. Board descriptions that sound right, but actual details show they do not actually have the control line to do the reset. You might think a 5 pin device sounds right but if the outputs are +5, +3.3, GND, RX, TX you can see you are missing the DTR/RTS or other serial handshake signal to initiate the bootloader.
2. Some board descriptions say 6 pin but looking at the pictures shows only 5 pins. BEWARE.
3. Right signals, but will not work with the O/S. This is typically a driver issue and you should get a driver update from the converter manufacturer or supplier.
4. You have everything hooked up, latest drivers loaded for you OS but the Arduino IDE cannot upload the sketch. This usually indicates that you have a board with a Chinese counterfeit USB to serial chip and the driver from the legitimate chip manufacturer recognizes that the part is “not right”. This is more common than you might think! The author has had to send a board to one of the bigger USB Serial chip manufacturers because of this issue.
5. You have the right drivers, the right chips, the right outputs but you cannot initiate the bootloader to do the sketch upload. Sometimes the reset signal out of the adapter needs to be appropriately “conditioned” for some board designs. DTR (RTS / RST) needs to go through a .1uf cap to accomplish reset. Fortunately the RF SnapIn has this cap on board so this step should not be required. For other boards, in order to be able to plug directly into an Arduino and be able to control the auto-program feature, you likely will need to make a modification either in your cable interconnect or on the board to add an in-line .1uf cap on the DTR (or RST) signal. It is possible that some boards that have an actual Arduino RST output instead of DTR or RTS and already have the in-line cap – you will need to check on your board. A Komby Reset Breakout is another way to solve this issue but is not required for the RF SnapIn board.
6. Serial port numbers are different on different boards. See the following section on configuring the IDE for the serial port number.
7. Didn’t install the serial adapter BEFORE running the IDE. In this case, you usually need to quit the IDE, install the adapter so the COM port is available and then restart the IDE to be able to see the needed COM port. Sometimes you get lucky and just exiting the serial selector and going back in gets the new adapter listed in the Arduino IDE serial pick list.

4.6.3 Adapter Recommendations:

The Komby FTDI Serial Programmer board was designed specifically to handle uploading sketches to the Arduino with a 5 pin (like on the RF SnapIn) and a 6 pin serial TTL sketch programming port found on some other boards. It can be obtained here: http://www.komby.com/ftdi_prog



At the same time, it might be a good idea to get the Komby Reset Breakout board. This allows non-Komby adapters to have the DTR, RTS, or RST signal conditioned to properly handle the reset on boards that need it. It can be obtained

here: <http://www.komby.com/reset>

We have also had good luck (i.e. works with Windows XP, 7 and 8) with a few different adapters from EBay / Alibaba / other sources:

“6 pin FTDI FT232RL USB to serial”



These are a bit more expensive than some of the other adapters, but then you get some data cable to play with rather than needing to supply a different way to get the signals to the board. As a bonus, these seem to use genuine FTDI parts thus work with the new drivers from FTDI including support for Win 8 and 8.1, etc. You may need to change around the wire order a bit to match the Komby 5 pin pin-out. We

use this one for all sketch updating (sometimes using a Komby reset board for other Komby board programming).



“USB 2.0 to TTL UART 6PIN Module Serial Converter CP2102”

This one does not use the FTDI chip, but rather the Silicon Labs part and seems to work well in all variations of Windows. The author has the first one pictured wired as a dedicated interface to a serial transmitter. It CANNOT be used to program sketches without modifications, or use of a manual reset so it is a little less convenient than the FTDI cable version. The one in the second picture was tested and DID NOT WORK for automatic programming of sketches as it has a RST to the CP2102 INPUT and 3.3v out instead of RTS/CTS or DTR as the first one has. Others might work, but again the key will be to have at least ONE flow control output pin that is used to trigger reset to enable the bootloader upload capability. An example is here:

<http://www.ebay.com/itm/6-Pin-USB-2-0-to-TTL-UART-Module-Serial-Converter-CP2102-STC-Replace-Module-WLSG-/201567625938>

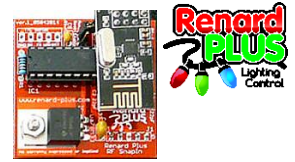
To reiterate: The key is to find an adapter with a 5 or 6 pin output that has one output as described above in the adapter requirements to enable auto-programming from the Arduino IDE. If you opt to go it yourself, you need to have good information about the boards. Use the pictures if the descriptions do not detail the pin outs. A custom interconnect cable MAY be required.

Example:



Note: Two pictured have an output that enables auto-program but one does not. One on left does have a bonus 3.3v so the NRF does not get wrong voltage if installed on the board during sketch uploads. However, “RST” is NOT actually RTS (request to send)!! This board would need serious modification to get RTS or DTR out to use with the auto-programming. The chip generates it, but the designer did NOT wire it out from the converter chip.

Note: watch for incorrect descriptions!! There are MANY offerings on EBay that say “6 pin” but the picture shows a 5 pin. Which is right- the description or the picture?!?! You should stay away from those just to be sure you get the right adapter.



4.7 Install the Arduino IDE

Installing the Arduino IDE is easy by getting the code from the Arduino site:

<http://arduino.cc/en/Main/Software>

You want the one that supports at least the “Arduino Uno” as that is what the board emulates. At the time this document was written, the stable release is 1.0.5 R2. Check the Komby Wiki and/or Github site to see what version is recommended for the Komby code.

The instructions on USING the IDE can be found here:

<http://arduino.cc/en/Guide/HomePage>

The instructions on downloading and installing the Komby receiver code can be found here:

<http://learn.komby.com/wiki/21/getting-started-guide>

4.8 Configure the serial port

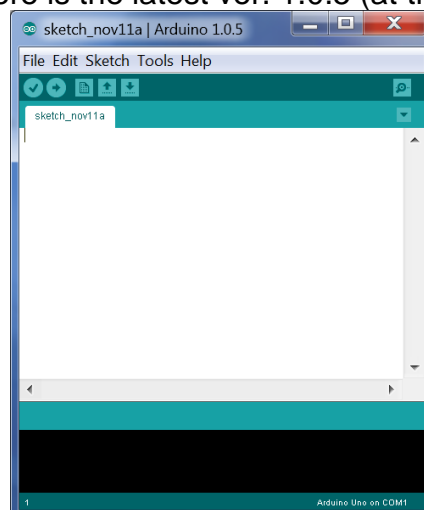
In order for the Arduino IDE to use the serial TTL adapter, you need to tell it what the port number of the device. This assumes your adapter has installed and you have the appropriate driver installed for it. If you plug it in and you do not get error messages, there is a good chance it is working.

4.8.1 Determine the COM Port

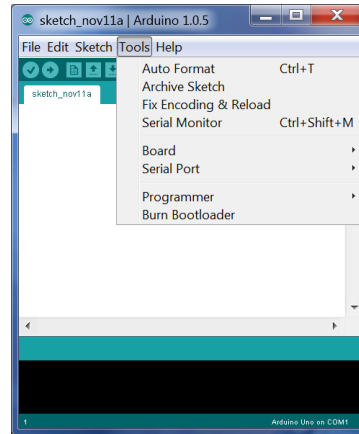
You can use the control panel Device Manager to determine which adapter has what serial port. With the adapter unplugged, open the “Device Manager” found in the Control Panel/System/Device Manager (or Hardware tab/Device Manager in XP) and then open up “Ports (COM & LPT)”. Note the existing com ports listed there with the adapter unplugged, then plug it in and see which port number is added. Note that number (and label that board with that number also). Usually the port number stays consistent to a specific board on a specific system. It may be a different port number on another computer – it just depends on what was installed before the port you are installing now.

4.8.2 Configure the IDE port

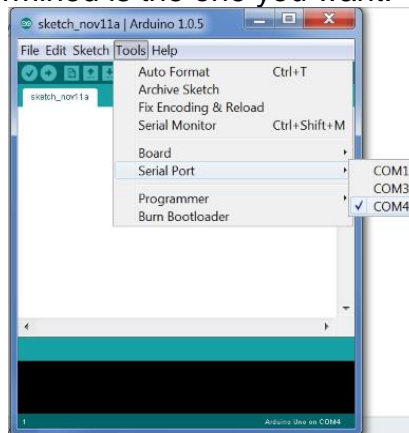
Now start up the IDE. Pictured here is the latest ver. 1.0.5 (at time of writing this doc) on Windows



Next select Tools on the toolbar:



Then select the com port you determined is the one you want:



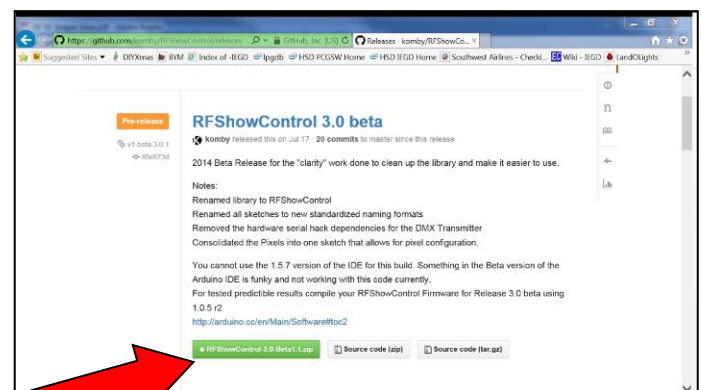
Now when you load the appropriate sketch, and select “upload”, the Arduino IDE will access the TTL Serial Adapter to upload to though the bootloader on the Atmega.

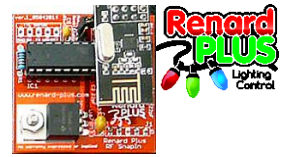
4.9 Komby code for RF SnapIn Receiver

You will want to use the Komby Github “RFShowControl” code and use the Receiver sketch making sure to set the target type to NRF1.

To get the most current copy of the release go to:
<https://github.com/komby/RFPixelControl/releases>

This will contain sketches for a wide variety of NRF based wireless applications including pixel control, strobes, etc. possible with the Komby boards. The RF-SnapIn will usually only require “RF_In_Renard_Out” or “RF_In_DMX_Out” depending on the firmware used in the main Renard Plus board in which the SnapIn is installed. The files names should be descriptive, or you can refer to the Komby Wiki for details.





The following are the configuration options you need to know about to tune your receiver if you are using Renard firmware on your controller: RF_In_Renard_out.ino

```
/****** START OF REQUIRED CONFIGURATION *****/
// NRF_TYPE Description: http://learn.komby.com/wiki/58/configuration-settings#NRF_TYPE
// Valid Values: RF1, MINIMALIST_SHIELD, WM_2999_NRF, RFCOLOR_2_4
#define NRF_TYPE RF1

// OVER_THE_AIR_CONFIG_ENABLE Description: http://learn.komby.com/wiki/58/configuration-
settings#OVER_THE_AIR_CONFIG_ENABLE
// Valid Values: OTA_ENABLED, OTA_DISABLED
#define OVER_THE_AIR_CONFIG_ENABLE 0

// RECEIVER_UNIQUE_ID Description: http://learn.komby.com/wiki/58/configuration-
settings#RECEIVER_UNIQUE_ID
// Valid Values: 1-255
#define RECEIVER_UNIQUE_ID 33
/****** END OF REQUIRED CONFIGURATION *****/

/****** START OF NON-OTA CONFIGURATION SECTION *****/
// LISTEN_CHANNEL Description: http://learn.komby.com/wiki/58/configuration-
settings#LISTEN_CHANNEL
// Valid Values: 0-83, 101-127 (Note: use of channels 84-100 is not allowed in the US)
#define LISTEN_CHANNEL 10

// DATA_RATE Description: http://learn.komby.com/wiki/58/configuration-settings#DATA_RATE
// Valid Values: RF24_250KBPS, RF24_1MBPS
#define DATA_RATE RF24_250KBPS

// HARDCODED_START_CHANNEL Description: http://learn.komby.com/wiki/58/configuration-
settings#HARDCODED_START_CHANNEL
// Valid Values: 1-512
#define HARDCODED_START_CHANNEL 1

// HARDCODED_NUM_CHANNELS Description: http://learn.komby.com/wiki/58/configuration-
settings#HARDCODED_NUM_CHANNELS
// Valid Values: 1-512
#define HARDCODED_NUM_CHANNELS 64

// RENARD_BAUD_RATE Description: http://learn.komby.com/wiki/58/configuration-
settings#RENARD_BAUD_RATE
// Valid Values: 19200, 38400, 57600, 115200, 230400, 460800
#define RENARD_BAUD_RATE 57600
/****** END OF NON-OTA CONFIGURATION SECTION *****/

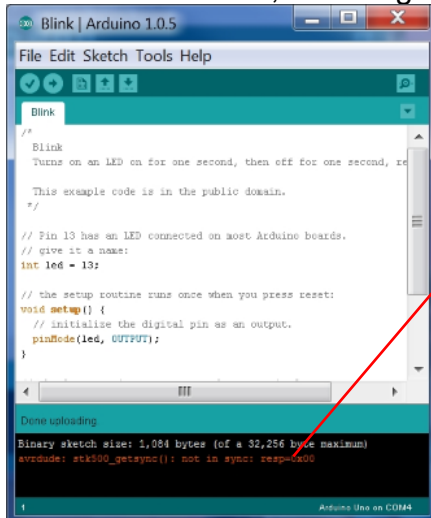
/****** START OF ADVANCED SETTINGS SECTION (OPTIONAL) *****/
// DEBUG Description: http://learn.komby.com/wiki/58/configuration-settings#DEBUG
// #define DEBUG 1

// FCC_RESTRICT Description: http://learn.komby.com/wiki/58/configuration-settings#FCC_RESTRICT
// Valid Values: 1, 0 (1 will prevent the use of channels that are not allowed in North America)
#define FCC_RESTRICT 1
/****** END OF ADVANCED SETTINGS SECTION *****/
```

Note: for DMX operation (Renard Plus programmed with DMX), please use the Komby Sketch "RF_IN_DMX_OUT.ino" in the release.

4.10 Program

Click Upload once your code tweaks are done and it should upload. Successful upload will result in no error messages so watch for errors- they are not always obvious! Connectivity errors (like serial driver issues, or wrong serial port selected) get an error during upload similar to:



Connectivity Error!!!
 avrdude:
 stk500_getsync(): not in
 sync: reapi=0x00

When you get a program success, you are ready to run!

4.11 RF Snapin Jumper Settings / Headers

4.11.1 Auto Reset Enable

This option allows for the USB Serial TTL programming board to reset the Atmega to go into Arduino Sketch Programming mode. Place a jumper on the jumper location when you will be updating the sketch, and remove it for normal receive operation.

4.11.2 RF SnapIn Header

This header is used to connect the RF SnapIn directly to the Renard Plus "Xbee Snap In" connector. The following is the pinout:

1	2	3	4	5
+5	N/C	GND	RX FROM 485	RX TO PIC

4.11.3 NRF Module Connector

If you are using Start Address Programming, you can use the PIC bypass to allow the data to flow thru the Renard Plus without the usual Renard "address eating". If you use a jumper across pins 1/2 then the data stream that comes into the device goes out exactly as it came in with no addresses consumed by the Renard Plus board. The default position is a jumper across pins 2/3.

Pin Layout

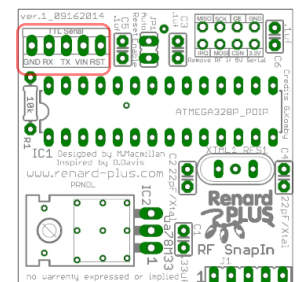
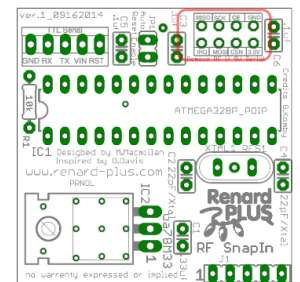
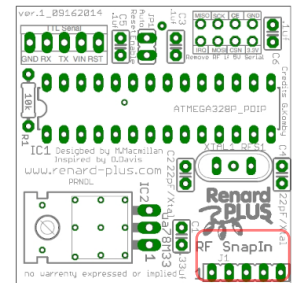
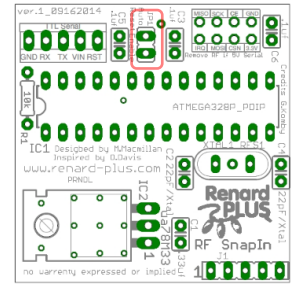
7	5	3	1
8	6	4	2

7-MISO 5-SCK 3-CE 1-GND
8-IRQ 6-MOSI 4-CS 2-Vcc

4.11.4 Serial TTL Arduino Programming Connector

This connector is used to program the Arduino sketches into the Atmega as described previously in the programming section.

1	2	3	4	5
GND	RX	TX	VIN	RST

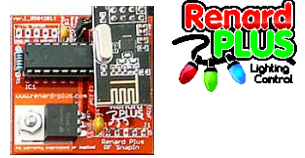


4.12 Computer Setup for RF Snapin

You will need to review the Transmitter setup for the type of wireless transmitter that you will use. The best is a Komby e1.31 setup (aka Komby “Sandwich”) that uses an Ethernet connection from the computer to the transmitter. See: <http://learn.komby.com/wiki/21/getting-started-guide> In the section “How do you want to get it into the air?” which describes the various transmitter options.

4.13 RF Snapin Final Testing

Final test is basically setting up the show computer to transmit with the controller ready to receive and sending out a test sequence. Alternatively, there is a “RF_Out_Rainbow_Test” sketch that can put a Komby board into transmit mode which puts out a full universe of fading colors. This is handy for testing the transmitter and receiver without involving the animation software and computer connections. Once you have an animation software controlling the controller via wireless, you are ready to deploy your setup. Have fun with Wireless Blinky!!



5. Notes

Use this page for YOUR notes about the boards.