# A Quick and Dirty RS232 WiFi Modem

# by TMA-1, teema1@yahoo.com

# Last update: December 29, 2019.

This document will help you build and/or use a WiFi modem that is compatible with your retro-computer's existing RS232 serial port.

**Background** (Feel free to skip this if it's old news to you.)

*So just what is a WiFi modem anyway?*

Back in the day, before the rise of the Internet and the Information Super-Highway, there were Bulletin Board Systems (BBS's) and the Telephone System. There were also "Hayes compatible" smart modems for accessing same. Smart, but not too fast. (Geek brownie points for the lowest baud rate you remember using.) They were also potentially expensive, as long distance still cost the moon and the stars at the time.

Alas, those days are gone forever. Or are they?

In 2016 some bright spark named Jussi Salim figured out that the ESP8266 (a cheap WiFi-enabled smart prototyping device similar to an Arduino) could be programmed to:

1. Sign on to your household WiFi to access the Internet, and
2. Telnet to Internet destinations by using simple commands patterned after the old industry-standard Hayes Smartmodem command set.

Another bright spark named Paul Richards made enhancements to Jussi's ESP8266 firmware code, enabling it to "answer" incoming "calls" and emulate for all intents and purposes the experience of placing and answering a call from a retro-computer to a BBS over the modem and telephone line.

Many SysOps of the past, (now somewhat older retro-computing enthusiasts like ourselves), have dusted off the floppy disk archive of the boards they took down years ago when the Internet displaced them all, and put them *back* up *on the Internet* using one of these WiFi devices.

Thanks to Jussi and Paul's work, all you need is a Commodore 64, *(down, Patrick)*, and one of these "modem emulators" for that machine, available commercially or as DIY.

*Commodore? What about the rest of us?* Ah-ha. That's where this document comes in!

Turns out the ESP8266 custom firmware also works with a different interface. A simplified version of the modem can be made with a cheap off-the-shelf part that gives the modem an RS232 connection to your favorite retro-box's existing RS232 serial port!

That is what this document is about. Complicated commercial RS232 WiFi Modems are available for about $55 or so plus shipping, but this one takes a minimalist "KISS" approach. You can build one yourself, with a minimum of time and effort, and usually for less than $10.

At the time of last update of this document, the design prototype has been tested with:

* Several flavors of 8088-based IBM-ish machines.
* A Pentium 4 class machine with a standard RS232 serial port.
* The TI-99/4A with a third party NanoPEB serial port (but with *warnings*, see below.)

The intent is also to test it soon with:

* The TI-99/4A with the "original" Peripheral Expansion Box (PEB) serial port.
* Macs, and an Apple II+ with its optional serial card.
* An Osborne One.
* Any other computer with an RS232 serial port, once *YOU* write me back and tell me how it went!

**Special Notes** - for TI-99/4A and **NanoPEB** users only:

There are several competing "designs" of NanoPEB, and the serial port has not been deployed consistently or even well among them. I can only speak to what worked for mine, and it was a struggle. (Mine is a recently-constructed NanoPEB with a TMS9902 UART based serial port.)

To get the NanoPEB serial port to work, the RTS (Ready To Send, pin 7) and CTS (Clear To Send, pin 8) of the RS232 must be shorted together, as documented in this forum thread:

[**https://atariage.com/forums/topic/299895-quest-for-terminal-software-that-works-with-the-quirky-nanopeb-serial-port/**](https://atariage.com/forums/topic/299895-quest-for-terminal-software-that-works-with-the-quirky-nanopeb-serial-port/)

It also requires hacked terminal emulator software, available in the first post of this thread:

[**https://atariage.com/forums/topic/300035-telco-patch-for-nanopeb-v1-serial/**](https://atariage.com/forums/topic/300035-telco-patch-for-nanopeb-v1-serial/)

A massive shout out to InsaneMultitasker for reverse engineering this and other telecomm packages to make them compatible with the NanoPEB's freaky RS232. Disassembled comment-less code is a bear!

**Also important: speed is currently limited to 2400 baud on this platform. Everywhere in this document where it says 9600 baud, *the TI-99/4A user must enter 2400 instead!***

*This is very much a work in progress*. I will try to keep this document updated if enhancements are made and implementation on this platform improves.

# Section 0 - If TMA-1 or ssshake Has Made One For You

Okay then. IF you were just handed this fat wad of paper, a funny looking modem, and possibly a 5 1/4” disk by a member of an unruly punk outfit named *Computer Heritage Group*, THEN this first **Section 0**" is the *only* one you need read. Similarly, if your name is Ed and your heart still aches for a dinosaur named Sue, you can do the same.

*If any part of the above paragraph* *confuses you*, then **skip immediately to Section 1** for complete detailed instructions on how to build and use your own RS232 WiFi modem!

Have your modem already then?

If you got a disk with it, it contains DOS terminal software for the PC. This copy of *Telix* is one retained from "back in the day", and is NOT tailored for WiFi use. It is already set up for 9600 baud, no parity, 8 data bits and 1 stop bit, software flow control.

The ESP8266 has already been set up with the required modem firmware. The non-volatile RAM has been largely set up, but you will need to update it with the SSID and password of YOUR particular home network.

Plug the modem into the Serial Port of your favorite retro-box. You will need a micro-USB cable to power the modem. Spark up Telix or the terminal program of your choice, set to 9600 baud, no parity, 8 data bits and 1 stop bit, software flow control.

* Hit [Enter] to get a blurb from the modem signifying that it is alive, and then enter:
* at$ssid=*yourssid*
* at$pass=*yourpassword*

You can check your work with at&v if you like.  
  
To connect to your WiFi enter:

* atc1

This may take a few attempts.

Now just one more thing. Update the ESP8266 non-volatile RAM to keep the SSID and password you've entered, by keying in:

* at&w

That's it! The ESP8266 will now connect automatically to your network every time it's powered on.

To connect to a BBS, enter something like:

# atdt particlesbbs.dyndns.org:6400 *or* atdt cavebbs.homeip.net

For help enter: atd?

See [**https://www.telnetbbsguide.com/bbs/list/brief/**](https://www.telnetbbsguide.com/bbs/list/brief/%20%20)  for a list of cool places to visit.

# Section 1 - DIY Materials

Right, you're going to build your own. Good on you.

Order the following from eBay or another favorite candy merchant:

* ESP8266 Prototyping Module: Do an eBay search for "ESP8266" and then sort by price. Should be just under $4.
* RS232 (female) to TTL adapter. Search for "RS232 TTL". Should be marginally over a dollar.
* Prototyping Circuit Board, 5x7 cm or thereabouts: On eBay, search for "Protoboard" and sort by price. You should find 5 cm by 7 cm pieces at 10 copies for $2-4. Share or keep the others--they're dead handy.
* Possibly required, DB25 female to DB9 male adaptor. The modem interface is DB9 female.
* Miscellaneous Materials: Wire, solder, hot glue.

That's about it folks. Just have to wait for the slow boat from China for the eBay stuff.

# Section 2 - Flashing the ESP8266 Firmware

* Download and install the drivers for flashing the ESP8266 module from:  
  [**http://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers**](http://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers)
* Download and install the firmware and flashing tool from: [**http://www.mediafire.com/file/tm71a1oa1a3macc/alwyz\_modded\_firmware.rar**](http://www.mediafire.com/file/tm71a1oa1a3macc/alwyz_modded_firmware.rar)
* Connect the module to your computer with a micro USB cable, and use Device Manager to figure out what logical serial port number the system has assigned to it.

Now to do the flash.

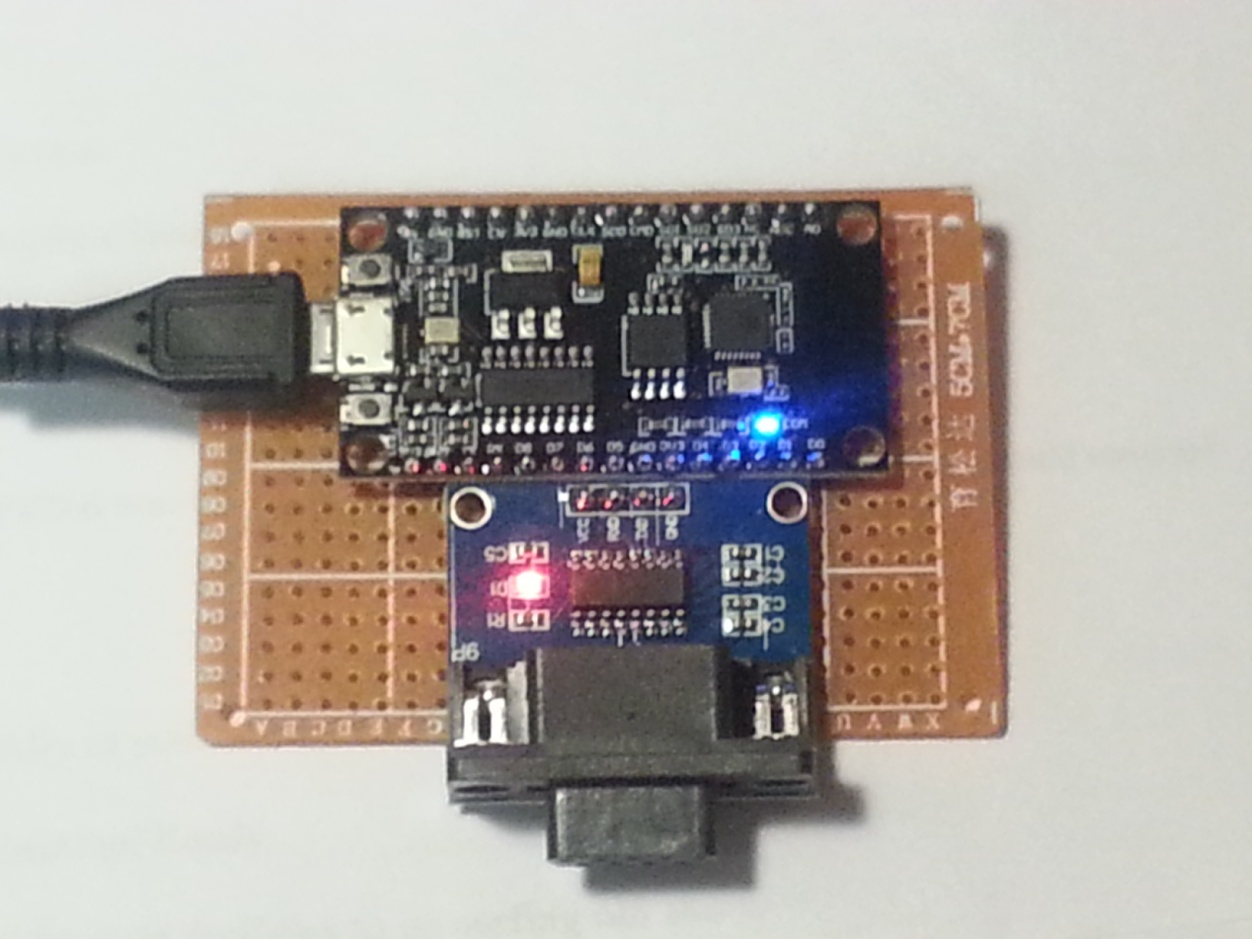
1. Open the flashing tool.
2. Fill in the COM port number where indicated.
3. Point the "Bin" selector to your copy of the **juno12.ino.generic.bin** file.
4. Now for the dexterity test. Press and hold the "Flash" button on the module, and,...
5. While holding "Flash", Press and release the "Reset" button on the module.
6. Click the "Download" button on your PC.

Once the download starts you can relax all your button fingers. When the download completes you may see a message that it "failed to leave download mode". This is benign.

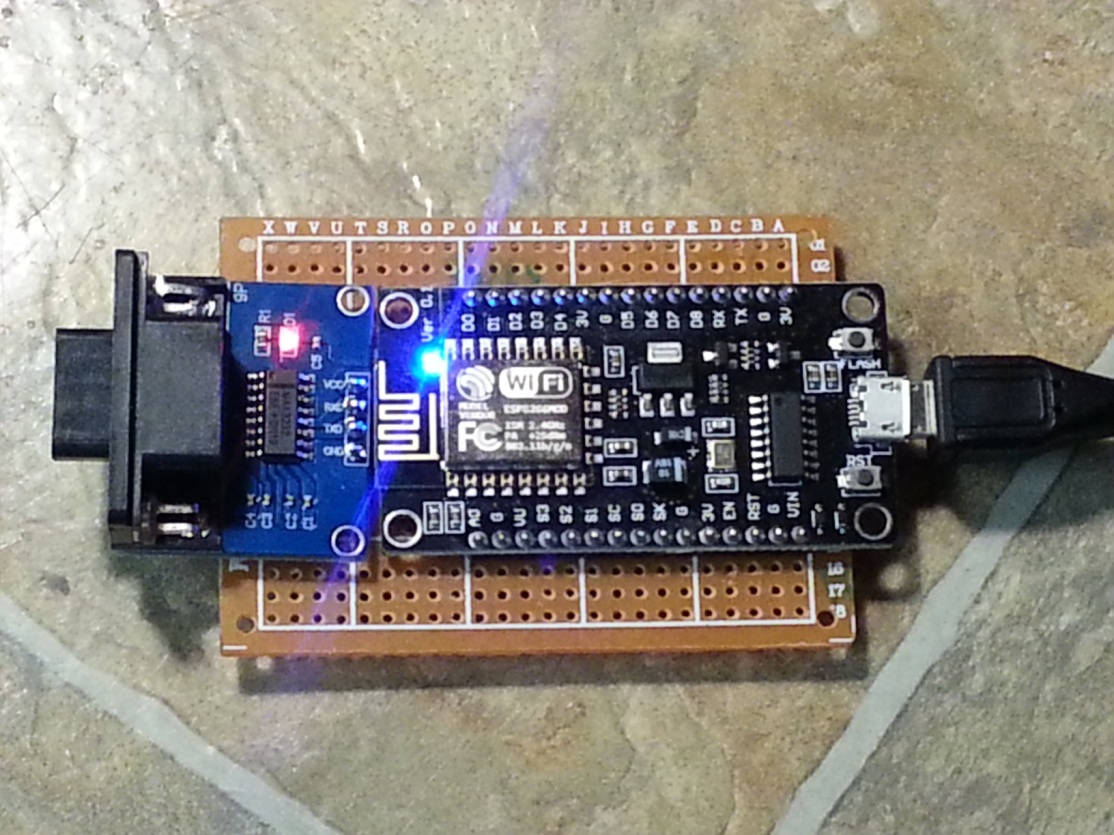
# Section 3 - Physical Construction

Lay out the two modules on the protoboard in some fashion that suits you and gives you access to connect the wires.

For my prototype, I unsoldered the 4 connection headers from the RS232-to-TTL adapter, and reversed them--mounting them on the underside of the board instead. Then I laid out the two modules more or less like this:



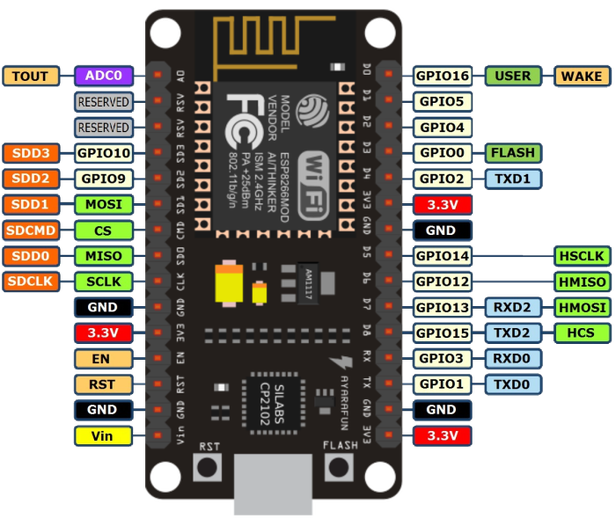
... or this:

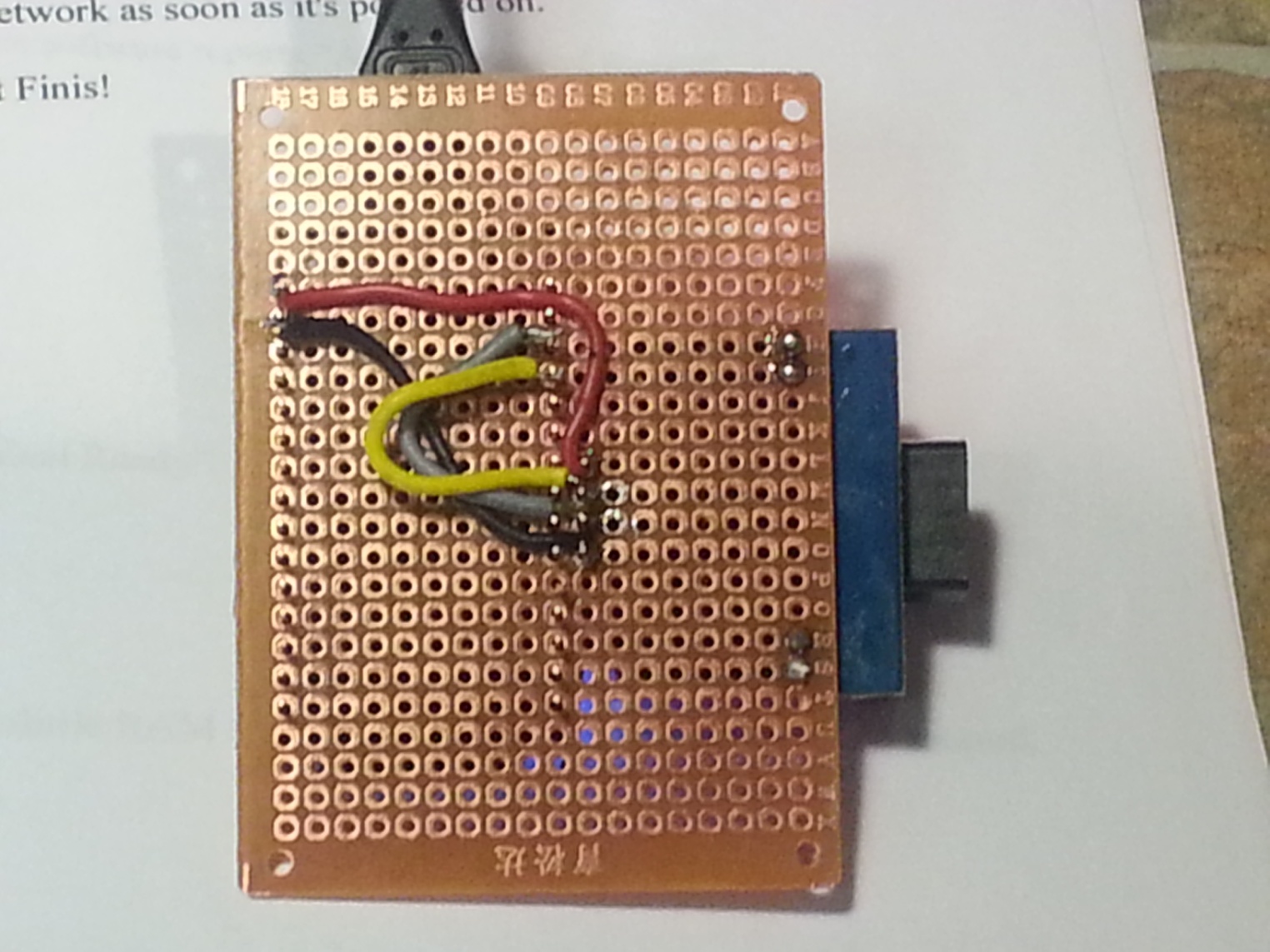


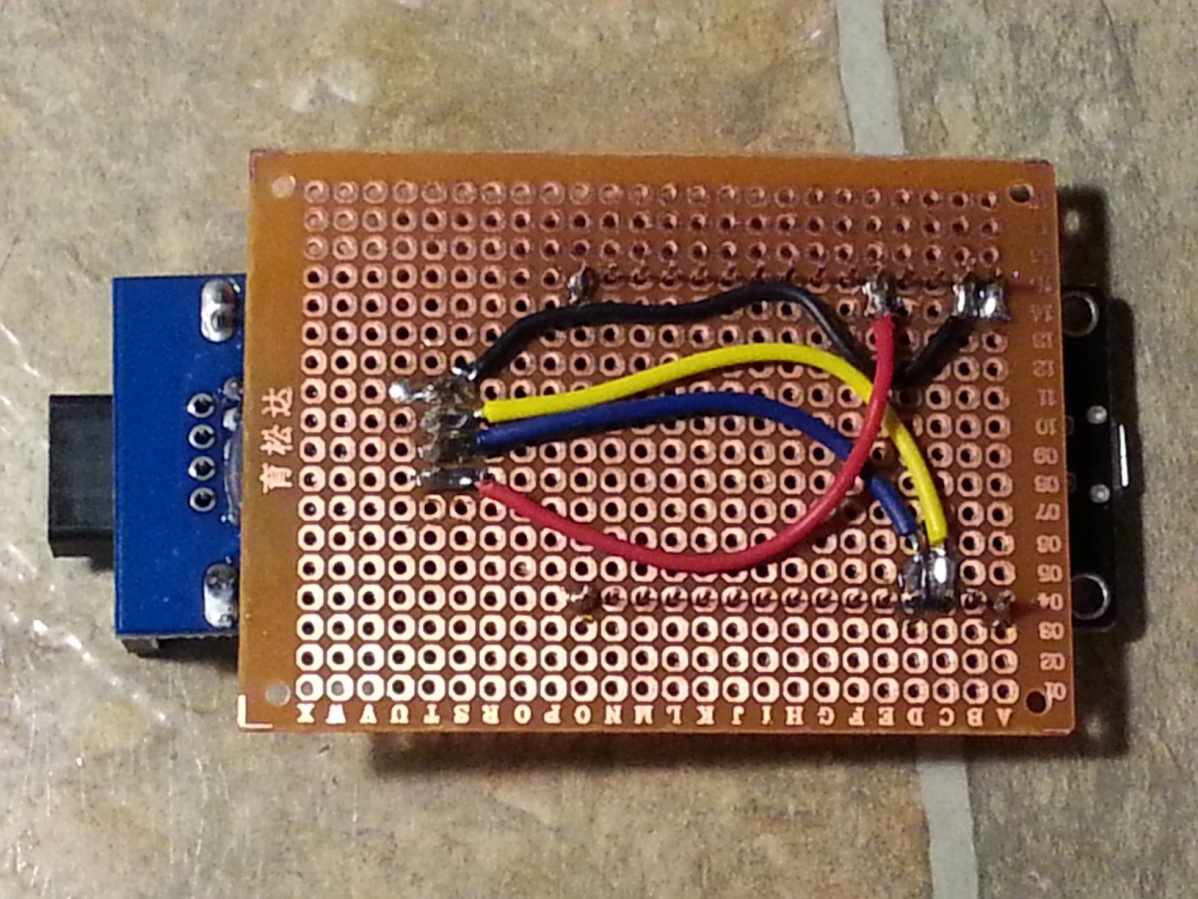
You may want to hot glue the RS232-to-TTL module down to the protoboard. I soldered the ground pins to the board instead. Your call. The ESP8266 tends to grip on its own with its many pins, but you can bend a couple of unused ones if you like, or solder them to the board.

Any way you mount it physically, you have to solder 4 wires as shown,

1. Vcc of the Serial-to-TTL module to a 3V pin of the ESP8222. (This is necessary to provide power to the MAX3232 chip on the RS232-to-serial board.)
2. Gnd of the Serial-to-TTL module to a Gnd of the ESP8222. (Big surprise.)
3. TXD of the Serial-to-TTL module to TXD0 of the ESP8222.  
     
   (Usually you connect *transmit* with *receive* and visa-versa, but the way my Serial-to-TTL is labelled, I found out the hard way that this wiring is the one that works. Or is the way my ESP8266 is labelled? Anyway, if it doesn't seem to respond when you're done, swap the wires.)
4. RCD of the Serial-to-TTL module to RXD0 of the ESP8222. (Surprised you again huh?)







That's about it for physical construction!

# Section 4 - First Use Only Steps

You are now ready to set the modem up for use with your household WiFi.

Plug the modem into the Serial Port of your favorite retro box, and light the fuse on your terminal program.

* Bring up a parameters menu. With Telix this is Alt-P.
* Change Baud Rate to "300" baud. (This is just the default of the flash image; don't worry, we won't stay here for long.)
* Use: N,8,1 (No parity, 8 data bits, and 1 stop bit.)
* Change the terminal type to ANSI-BBS or equivalent.
* Set flow control to "Software" or "XON/OFF" control.
* After exiting the parameters set-up, hit [Enter] to get a blurb from the modem software signifying that it is alive and kicking.

(If you get no response at this point, check:

* Is power getting to both the ESP8266 and the RS232 interface module?
* Is the Tx connected to TX, and Rx to Rx?
* Are the communication parameters for the active session set correctly in the software?)

Now enter the following two commands to set the SSID and password for your wireless:

* at$ssid=*yourssid*
* at$pass=*yourpassword*

You can check your work with at&v if you like.

Now enter the following to connect:

* atc1

This may take a few attempts. You are now connected to your WiFi, but we're not *quite* done yet,...

Enter the following commands to alter the defaults, setting software flow control on, and changing the modem parameters to 9600 baud:

* at&p0
* at&k2
* at$sb=9600

Now we have to update our terminal parameters to match.

* Call up your parameters menu again, and select 9600 Baud

If you're using Telix, make sure to set and save your parameter defaults with Alt-O for next time.

Now just one more very important thing. Update the ESP8266 non-volatile RAM to keep the SSID, password, and other parameters you've entered, by entering:

* at&w

That's it. The ESP8266 will now connect automatically to your network every time it's powered on.

To connect to a BBS, enter something like:

# atdt particlesbbs.dyndns.org:6400 *or* atdt cavebbs.homeip.net

For help enter: adt?

See [**https://www.telnetbbsguide.com/bbs/list/brief/**](https://www.telnetbbsguide.com/bbs/list/brief/%20%20)  for a list of cool places to visit.

C'est Finis!