Retro WiFi modem

Introduction

This is my version of the RetroWiFiModem. All I have done is modify the PCB layout to use a Node MCU development board instead of the original Wemos D1 Mini

The code was also modified to include Node MCU support

Important Note

Please note that there are 2 standards (pin-outs) for the 10 pin motherboard connector
AT/Everex or Intel/DTK
This modem is using the Intel/DTK pinout
Therefore be aware of this when buying/making your connections to the modem serial port

Post Build Setup

The default serial configuration is 1200bps, 8 data bits, no parity, 1 stop bit.

1. `AT$SSID=your WiFi network name` to set the WiFi network that the modem will connect to when it powers up.
2. `AT$PASS=your WiFi network password` to set the password for the network.
3. `ATC1` to connect to the network.
4. Optional stuff:
   * `AT$SB=speed` to set the default serial speed.
   * `AT$SU=dps` to set the data bits, parity and stop bits.
   * `ATNETn` to select whether or not to use Telnet protocol.
   * `AT&Kn` to use RTS/CTS flow control or not.
5. `AT&W` to save the settings.

Once you've done that, the modem will automatically connect to your WiFi network on power up and will be ready to "dial up" a connection with ATDT.
Command Reference
Multiple AT commands can be typed in on a single line. Spaces between commands are allowed, but not within commands (i.e. AT S0=1 X1 Q0 is fine; ATS 0= 1 is not). Commands that take a string as an argument (e.g. AT$SSID=, AT$TTY=) assume that *everything* that follows is a part of the string, so no commands are allowed after them.

Command | Details

+++	Online escape code. Once your modem is connected to another device, the only command it recognises is an escape code of a one second pause followed by three typed plus signs and another one second pause, which puts the modem back into local command mode.
A/	Repeats the last command entered. Do not type AT or press Enter.
AT	The attention prefix that precedes all command except A/ and +++.
AT?	Displays a help cheatsheet.
ATA	Force the modem to answer an incoming connection when the conditions for auto answer have not been satisfied.
ATC? ATC*n*	Query or change the current WiFi connection status. A result of 0 means that the modem is not connected to WiFi, 1 means the modem is connected. The command ATC0 disconnects the modem from a WiFi connection. ATC1 connects the modem to the WiFi.
ATDS*n*	Calls the host specified in speed dial slot *n* (0-9).
ATDT<i>[+=-]host[:port]	Tries to establish a WiFi TCP connection to the specified host name or IP address. If no port number is given, 23 (Telnet) is assumed. You can also use ATDT to dial one of the speed dial slots in one of two ways: - The alias in each speed dial slot is checked to see if it matches the specified hostname. - A host specified as 7 identical digits dials the slot indicated by the digit. (i.e. 2222222 would speed dial the host in slot 2). Preceding the host name or IP address with a +, = or - character overrides the ATNET setting for the period of the connection. **+** forces NET2 (fake Telnet) **=** forces NET1 (real Telnet) **-** forces NET0 (no Telnet) Once the dial attempt has begun, pressing any key before the connection is established will abort the attempt.
ATE? ATE*n*	Command mode echo. Enables or disables the display of your typed commands. - E0 Command mode echo OFF. Your typing will not appear on the screen. - E1 Command mode echo ON. Your typing will appear on the screen.

ATGET*http://host[/page]*	Displays the contents of a website page. **https** connections are not supported. Once the contents have been displayed, the connection will automatically terminate.
ATH	Hangs up (ends) the current connection.
ATI	Displays the current network status, including sketch build date, WiFi and call connection state, SSID name, IP address, and bytes transferred.
ATNET? ATNET*n*	Query or change whether telnet protocol is enabled. A result of 0 means that telnet protocol is disabled; 1 is *Real* telnet protocol and 2 is *Fake* telnet protocol. If you are connecting to a telnet server, it may expect the modem to respond to various telnet commands, such as terminal name (set with `AT$TTY`), terminal window size (set with `AT$TTS`) or terminal speed. Telnet protocol should be enabled for these sites, or you will at best see occasional garbage characters on your screen, or at worst the connection may fail. The difference between *real* and *fake* telnet protocol is this: with *real* telnet protocol, a carriage return (CR) character being sent from the modem to the telnet server always has a NUL character added after it. The implementation of the telnet protocol used by some BBSes doesn't properly strip out the NUL character. When connecting to such BBSes (Particles! is one), use *fake* telnet. When using *real* telnet protocol, when the telnet server sends a CR character followed by a NUL character, only the CR character will be sent to the serial port; the NUL character will be silently stripped out. With *fake* telnet protocol, the NUL will be passed through.
ATO	Return online. Use with the escape code (+++) to toggle between command and online modes.
ATQ? ATQ*n*	Enable or disable the display of result codes. The default is Q0. Q0 Display result codes Q1 Suppress result codes (quiet).
ATRD ATRT	Displays the current UTC date and time from NIST in the format *YY-MM-DD HH:MM:SS*. A WiFi connection is required and you cannot be connected to another site.
ATS0? ATS0=*n*	Display or set the number of "rings" before answering an incoming connection. Setting `S0=0` means "don't answer".
ATV?	Display result codes in words or numbers. The default is V1.

ATV*n*	V0 Display result codes in numeric form. V1 Display result codes in text form.
ATX? ATX*n*	Control the amount of information displayed in the result codes. The default is X1 (extended codes). Display basic codes (CONNECT, NO CARRIER). X1 Display extended codes (CONNECT speed, NO CARRIER (connect time))
ATZ	Resets the modem.
AT&F	Reset the NVRAM contents and current settings to the sketch defaults. All settings, including SSID name, password and speed dial slots are affected.
AT&K? AT&K*n*	Data flow control. Prevents the modem's buffers for received and transmitted from overflowing. &K0 Disable data flow control. &K1 Use hardware flow control. Requires that your computer and software support Clear to Send (CTS) and Request to Send (RTS) at the RS-232 interface.
AT&R? AT&R=*server pwd*	Query or change the password for incoming connections. If set, the user has 3 chances in 60 seconds to enter the correct password or the modem will end the connection.
AT&V*n*	Display current or stored settings. &V0 Display current settings. &V1 Display stored settings.
AT&W	Save current settings to NVRAM.

AT&Zn?
AT&Z*n*=*host[:port],alias*
	Store up to 10 numbers in NVRAM, where *n* is the position 0-9 in NVRAM, and *host[:port]* is the host string, and *alias* is the speed dial alias name. The host string may be up to 50 characters long, and the alias string may be up to 16 characters long. Example:
	`AT&Z2=particlesbbs.dyndns.org:6400,particles` This number can then be dialed in any of the following ways:
	`ATDS2`
	`ATDTparticles`
	`ATDT2222222`

AT$AE? AT$AE=*startup AT cmd*	Query or change the command line to be executed when the modem starts up.
AT$BM? AT$BM=*server busy msg*	Query or change a message to be returned to an incoming connection if the modem is busy (i.e. already has a connection established).
AT$MDNS AT$MDNS=*mDNS name*	Query or change the mDNS network name (defaults to "espmodem"). When a non zero TCP port is defined, you can telnet to that port with **telnet mdnsname.local port**.

AT$PASS? AT$PASS=*WiFi pwd*	Query or change the current WiFi password. The password is case sensitive. Clear the password by issuing the set command with no password. The maximum length of the password is 64 characters.
AT$SB? AT$SB=*n*	Query or change the current baud rate. Valid values for "n" are 110, 300, 450, 600, 710, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 and 115200. Any other value will return an ERROR message. The default baud rate is 1200. The Retro WiFi modem does not automatically detect baud rate like a dial-up modem. The baud rate setting must match that of your terminal to operate properly. It will display garbage in your terminal otherwise.
AT$SP? AT$SP=*n*	TCP server port to listen on. A value of 0 means that the TCP server is disabled, and no incoming connections are allowed.
AT$SSID? AT$SSID=*ssid name*	Query or change the current SSID to the specified name. The given SSID name is case sensitive. Clear the SSID by issuing the set command with no SSID. The maximum length of the SSID name is 32 characters.
AT$SU? AT$SU=*dps*	Query or change the current number of data bits ('d'), parity ('p') and stop bits ('s") of the serial UART. Valid values for 'd' are 5, 6, 7 or 8 bits. Valid values for 'p' are (N)one, (O)dd or (E)ven parity. Valid values for 's' are 1 or 2 bits. The default settings are 8N1. The UART setting must match your terminal to work properly.
AT$TTL? AT$TTL=*telnet location*	Query or change the Telnet location value to be returned when the Telnet server issues a SEND-LOCATION request. The default value is "Computer Room".
AT$TTS? AT$TTS=*WxH*	Query or change the window size (columns x rows) to be returned when the Telnet server issues a NAWS (Negotiate About Window Size) request. The default value is 80x24. For terminals that are smaller than 80x24, setting these values appropriately will enable paging on the help (AT?) and network status (ATI) commands.
AT$TTY? AT$TTY=*terminal type*	Query or change the terminal type to be returned when the Telnet server issues a TERMINAL-TYPE request. The default value is "ansi".
AT$W? AT$W=*n*	Startup wait $W=0 Startup with no wait. $W=1 Wait for the return key to be pressed at startup.

Updating the Software

The software may be loaded in the normal way or you may use the default OTA upload capability built into the Arduino IDE to upload OTA

References

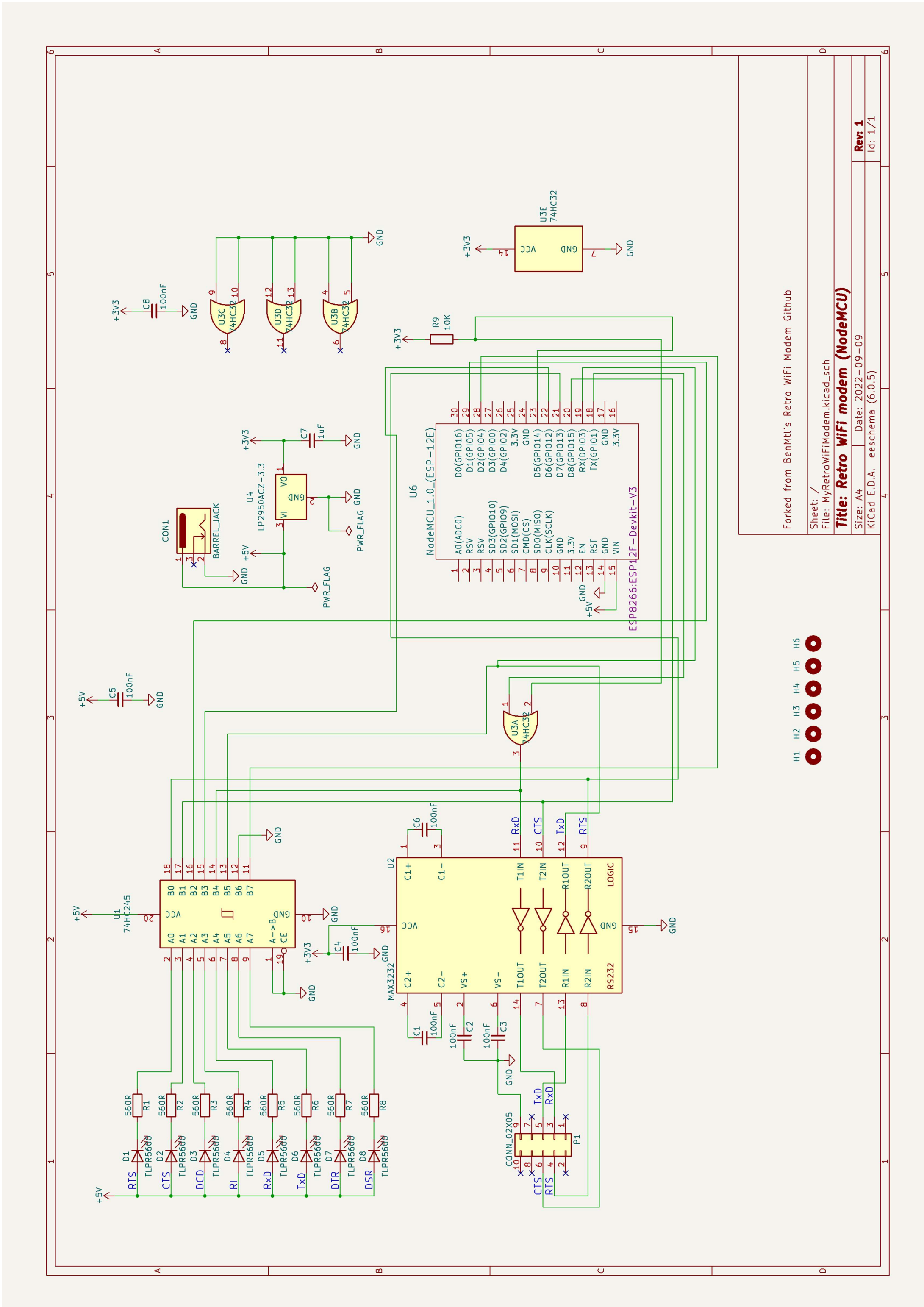
* WiFi232 - An Internet Hayes Modem for your Retro computer
http://biosrhythm.com/?page_id=1453

- WiFi232's Evil Clone (https://forum.vcfed.org/index.php?threads/wifi232s-evil-clone.1070412/)
- Jussi Salin's Virtual modem for ESP8266 (https://github.com/jsalin/esp8266_modem)
- Stardot's ESP8266 based virtual modem (https://github.com/stardot/esp8266_modem)
- Roland Juno's ESP8266 based virtual modem (https://github.com/RolandJuno/esp8266_modem)

Acknowledgements

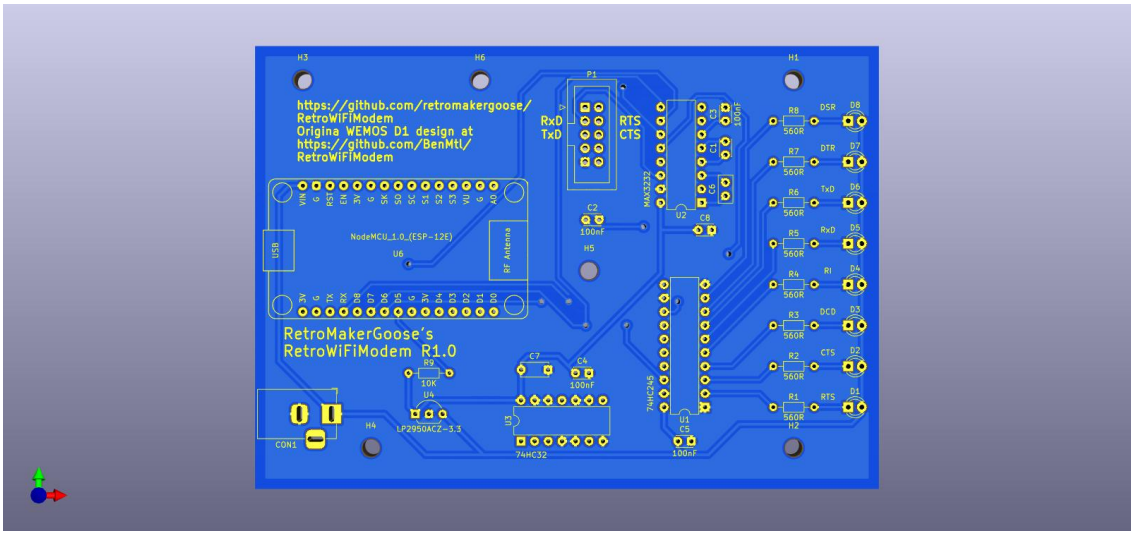
- A whole lot of people owe a big vote of thanks to Jussi Salin for releasing their virtual modem code for the ESP8266 and starting the ball rolling.
- Paul Rickards for an amazing bit of hardware to draw inspiration from.
- All the Stardot contributors for their work.
- And, of course, Dennis C. Hayes for creating something so simple and elegant that has stood the test of time.

Schematic

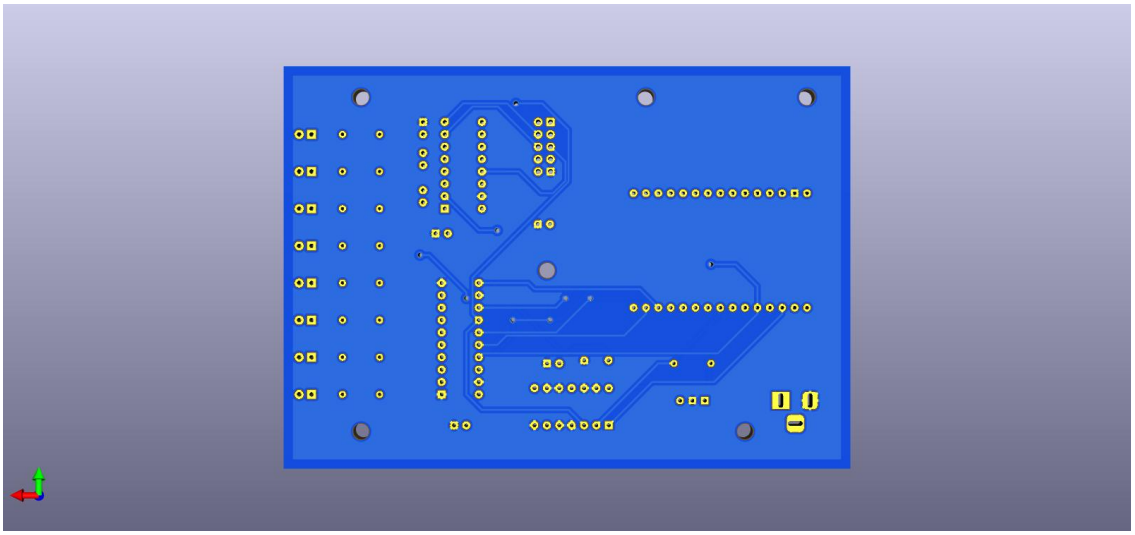

Forked from BenMtl's Retro WiFi Modem Github
Sheet: /
File: MyRetroWiFiModem.kicad_sch
Title: Retro WiFi modem (NodeMCU)
Size: A4
Date: 2022-09-09
Rev: 1
KiCad E.D.A. eeschema (6.0.5)
Id: 1/1

PCB

Front



Back



BOM

C7,	1	1uF	Unpolarized capacitor, small symbol
CON1,	1	BARREL_JACK	DC Barrel Jack with an internal switch
D1, D2, D3, D4, D5, D6, D7, D8,	8	TLPR5600	Light emitting diode
H1, H2, H3, H4, H5, H6,	6	MountingHole	Mounting Hole without connection
P1,	1	CONN_02X05	Generic connector, double row, 02x05, odd/even pin numbering scheme (row 1 odd numbers, row 2 even numbers), script generated (kicad-library-utils/schlib/autogen/connector/)
R1, R2, R3, R4, R5, R6, R7, R8,	8	560R	Resistor
R9,	1	10K	Resistor
U1,	1	74HC245	Octal BUS Transceivers, 3-State outputs
U2,	1	MAX3232	Dual RS232 driver/receiver, 5V supply, 120kb/s, 0C-70C
U3,	1	74HC32	Quad 2-input OR
U4,	1	LP2950ACZ-3.3	Positive 100mA 30V Linear Micropower Voltage Regulator, Fixed Output 3.3V, TO-92
U6,	1	NodeMCU_1.0_(ESP-12E)	