

HUZZAH ESP8266 WiFi Modem Setup

© 2017 Chris Schneider (SHIFT838)

Version 1.0



Last Updated: 07/06/2017

Table of Contents

1. Acknowledgements	4
2. Project Information	4
3. Requirements	5
3.1 Hardware Requirements	Error! Bookmark not defined.
3.2 Software Requirements	6
4. The Code	6
4.1 Download the Code	6
5. Enclosure & Component Preparation	7
5.1 Enclosure	7
5.2 Components	7
6. Modifying the Code	9
7. Connecting the FTDI programming cable	10
8. Setting up Software for Use with HUZZAH ESP8266 WiFi Board	11
9. Programming HUZZAH ESP8266 WiFi Board	15
10. Wiring Diagrams	18
10.1 TI-WiFi Component / Cable Wiring	18
10.2 Cable Wiring Pinout	19
11. Powering up the Unit	20
11.1 Connect Cables	20
11.2 First Time Boot	20
12. Command Set Summary	22
12.1 Help	22
12.2 Auto Answer	22
12.3 Command Mode	22
12.4 Dial Host	22
12.5 Echo	22
12.6 Exit Command Mode	23
12.7 Factory Defaults	23
12.8 Flow Control	23
12.9 Hangup/Disconnect	23

12.10	HTTP GET.....	23
12.11	Load NVRAM.....	24
12.12	Network Info.....	24
12.13	PETSCII	24
12.14	Pin Polarity	24
12.15	Query Commands.....	25
12.16	Save to NVRAM	25
12.17	Set Baud Rate	25
12.18	Set Busy Message	25
12.19	Set Speed Dial Entries	26
12.20	Set Server Listening Port	26
12.21	Set WiFi Password	26
12.22	Set WiFi SSID.....	26
12.23	Show Settings	26
12.24	Speed Dialing Stored Entries.....	27
12.25	Telnet Command Handling	27
12.26	Verbose Mode	27
12.27	WiFi (Connect/Disconnect)	27
13.	TI/Geneve BBS (via Telnet)	28
14.	Support	28

1. Acknowledgements

The firmware used for this project was developed by Paul Rickards in 201 and has been released under the GPLv3 license. The firmware was based on the original code by Jussi Salin in 2015 and released under the GPLv3 license.

This project is not to compete with Paul's retail release WiFi232 modem which is superior to this project as it offers other options this does not. This project is simply to show how I got it all working within his code and for any DIY'ers out there that would like to try.

Many thanks go out to Paul for being very responsive and answering my many questions to understand how this thing worked.

Paul Rickards code that will be used for this project is located at:

https://github.com/RolandJuno/esp8266_modem

Jussi Salin's original code is located at:

https://github.com/jsalin/esp8266_modem

2. Project Information

This project will take a HUZZAH ESP8266 WiFi microcontroller and pair it with a RS232 to TTL converter board along with a USB breakout board for powering the unit. A LED, power switch and 2 x momentary switches will be added for added functionality.

The HUZZAH EPS8266 is a Wi-Fi enabled microcontroller that can add Internet to your next project. It is an 80 MHz microcontroller with a full Wi-Fi front-end (both as client and access point) and TCP/IP stack with DNS support as well. While this chip has been very popular, it has also been very difficult to use.

I have been successful getting the device coded with Paul Rickards firmware that is very familiar to many of us. These commands are based on or a very familiar Hayes AT command set. (See Acknowledgements section).

3. Requirements

3.1 Skill Requirements

- Some basic solder skills
- Basic electronic understanding
- Patience

3.2 Hardware Requirements

- FTDI Cable to update the firmware
https://www.amazon.com/gp/product/B00QT7LQ88/ref=oh_aui_detailpage_o06_s00?ie=UTF8&psc=1

Note: I have found a couple of cables and or FTDI boards that will not code the ESP8266 for some reason or another and I always get an Error 1 when trying to code. But this little \$6.00 FTDI cable has never failed me.

- HUZZAH ESP8266 Breakout Board
https://www.amazon.com/gp/product/B01MDSUKHE/ref=oh_aui_detailpage_o09_s00?ie=UTF8&psc=1
- RS232 to TTL Converter with Male DB9 (3.3v to 5v)
https://www.amazon.com/gp/product/B00OPU2QJ4/ref=oh_aui_detailpage_o04_s00?ie=UTF8&psc=1
- USB Type A Female Breakout board (I use this to power the ESP8266)
https://www.amazon.com/Female-Breakout-Board-2-54mm-Header/dp/B01K42V2S2/ref=sr_1_8?s=electronics&ie=UTF8&qid=1492840310&sr=1-8&keywords=usb+breakout

- Decent pencil type soldering iron and
- Thin Sn63/Pb37 solder
- 1 x DSUB 9 Female
- 1 x DSUB25 Male
- 8 Conductor wire
- USB Cable for power
- Project box to put it all in
- 2.54mm Male headers (I used a total of 18)
- 10 x Female to Female jumper wires

- 2 x Momentary push button switches (normally open)
- 1 x SPST micro rocker switch
- 1 x 5mm Blue LED and Bezel to fit
- 1 x 120 Ohm Resistor ¼ watt

3.3 Software Requirements

- Arduino IDE 1.8.0 or higher installed
<https://www.arduino.cc/en/Main/Software>
- ESP8266 Libraries for Arduino IDE
http://arduino.esp8266.com/stable/package_esp8266com_index.json
This is configured within Arduino IDE program for Additional Boards
Manager URL's under **File → Preferences**

4. The Code

4.1 Download the Code

The below links is to the firmware I used that has various AT commands coded in order to act like a Hayes modem. The code has an option to hold down GPIO0 for at least 5 seconds and when released the device will boot into 300 baud. This is useful if for some reason you cannot access it on a computer that does not operate at the configured baudrate that is hard coded in the VRAM.

For example if you want to move it to a Ti-99/4A without the F18A and it was previously used on a F18A enabled system that supported the higher baudrate and the unit was hardcoded in NVRAM with a higher baudrate such as 9600 or above.

Firmware Code:

https://github.com/RolandJuno/esp8266_modem

5. Enclosure & Component Preparation

5.1 Enclosure

If you have not decided the best enclosure to use then this is the time. I used a small aluminum enclosure as it's easy to work with. I have heard of aluminum enclosures causing signal issues with the wireless antenna, but I have not had the issue. Of course my unit sits about 3 feet from my wireless router too.

It is best at this point to decide how the best placement of each component within the enclosure is.

- Create standoffs if needed
 - I used some plastic screws/bolts with nuts. Cheap and easy.
- Cut appropriate holes required to mount components such as:
 - Momentary switches
 - LED & Bezel
 - Rocker power switch
 - DB9 Serial to TTL adapter board
 - USB connector

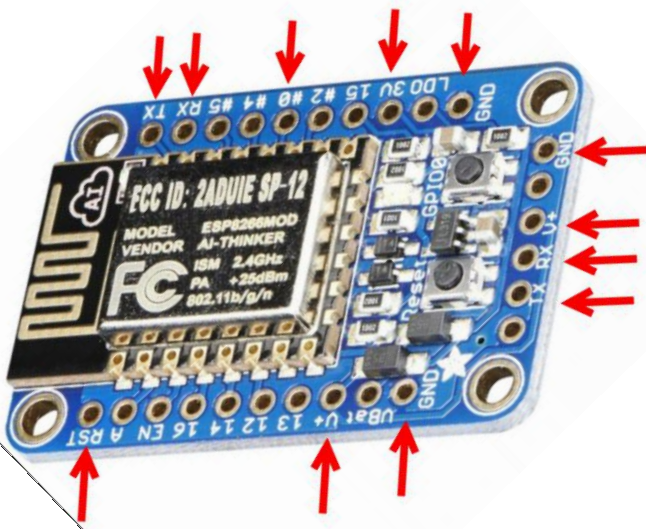
5.2 Components

It is best at this point to solder the header connectors to the boards.

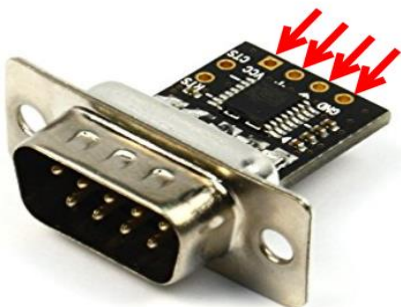
After determining the best way to mount the components you will now know the best orientation for the header connectors, whether it will be straight or right angled headers on top or bottom based on the enclosure chosen and how your components will fit within it.

The headers I soldered were as annotated below:

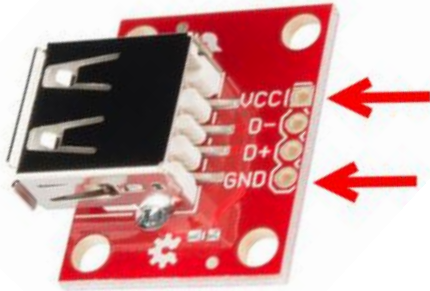
HUZZAH ESP8266:

	<p>Headers:</p> <p>Top: RX, TX, #0, 3V, GND</p> <p>Side: GND, V+, RX, TX</p> <p>Bottom: GND, V+ RST</p> <p>Note: Side connections are for programming with FTDI cable.</p>
--	---

RS232 to TTL Converter:

	<p>Headers:</p> <p>GND, RX, TX, VCC</p> <p>Picture is for reference only. There are a ton of manufactures of this type of unit and yours may not look exactly like this. Just match up the pins to what you purchase.</p>
---	---

USB Type A Female Breakout Board:

	<p>Headers:</p> <p>GND, VBUS/VCC</p> <p>Only used for powering the unit.</p> <p>VBUS & VCC are just voltage in. Depending on the board purchased it will have one of the two labels.</p> <p>Picture is for reference only. There are a ton of manufactures of this type of unit and yours may not look exactly like this. But the pinout should be the same.</p>
---	--

6. Modifying the Code

There is one command that I found that is remarked out but works and if you want it to display within the help and current/stored settings areas it's easy to accomplish. This command is setting the listen port of your device. Default is 6400, I changed mine to 9900. Leaving it at 6400 is just fine. This update is only if you want to see it displayed on the help and settings screens.

If you want to see the extra command when issuing the **AT?** Command then find the code areas that are commented out with `/// and remove the /// before the code.`

Search for:

```
//Serial.print("Server TCP PORT: ")
```

Should be two entries you find. One under each of the below code headings:

```
void displayCurrentSettings()
```

void displayStoredSettings()

Now search for:

```
//Serial.println("SERVER PORT....");
```

Should be under the below display heading

void displayHelp()

The above code snippets is just the beginning of the line of code. Just remove the ‘//’ from the beginning of the lines and save.

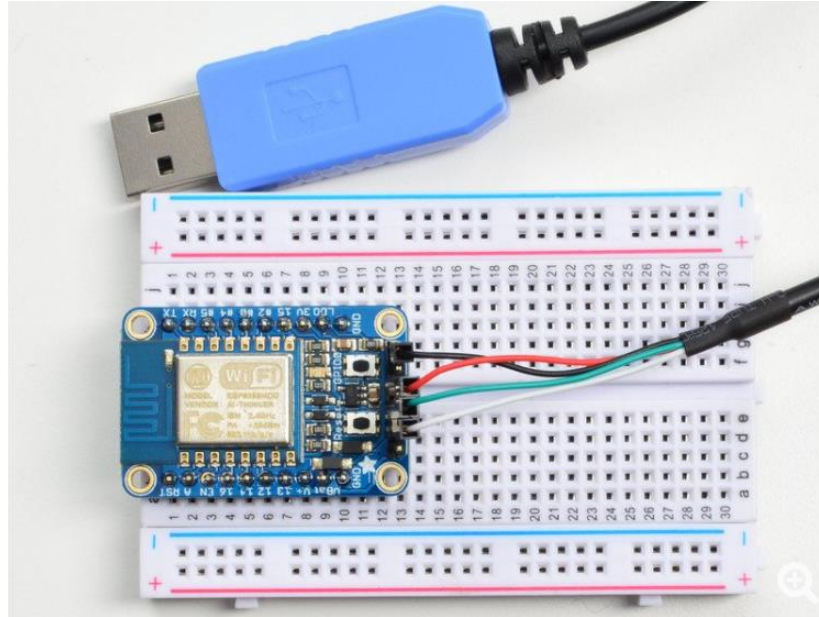
7. Connecting the FTDI programming cable

To hook up the FTDI cable to your PC is simple, just plug it in. Verify on your computer which serial port has been assigned to the FTDI. You will need to hook it up to the HUZZAH ESP8266 to program upload the firmware to the device. My FTDI cable is referenced below but be sure to verify what connectors go where for your FTDI / Console cable.

I also recommend if using one of these since the wires can be frail to use 4 male to female header extension cables on the ends.

My FTDI cable was coded as below, but please verify yours is correct.

Color	Pin
Black	Ground
Red	V+
Green	RX
White	TX



8. Setting up Software for Use with HUZAZH ESP8266 WiFi Board

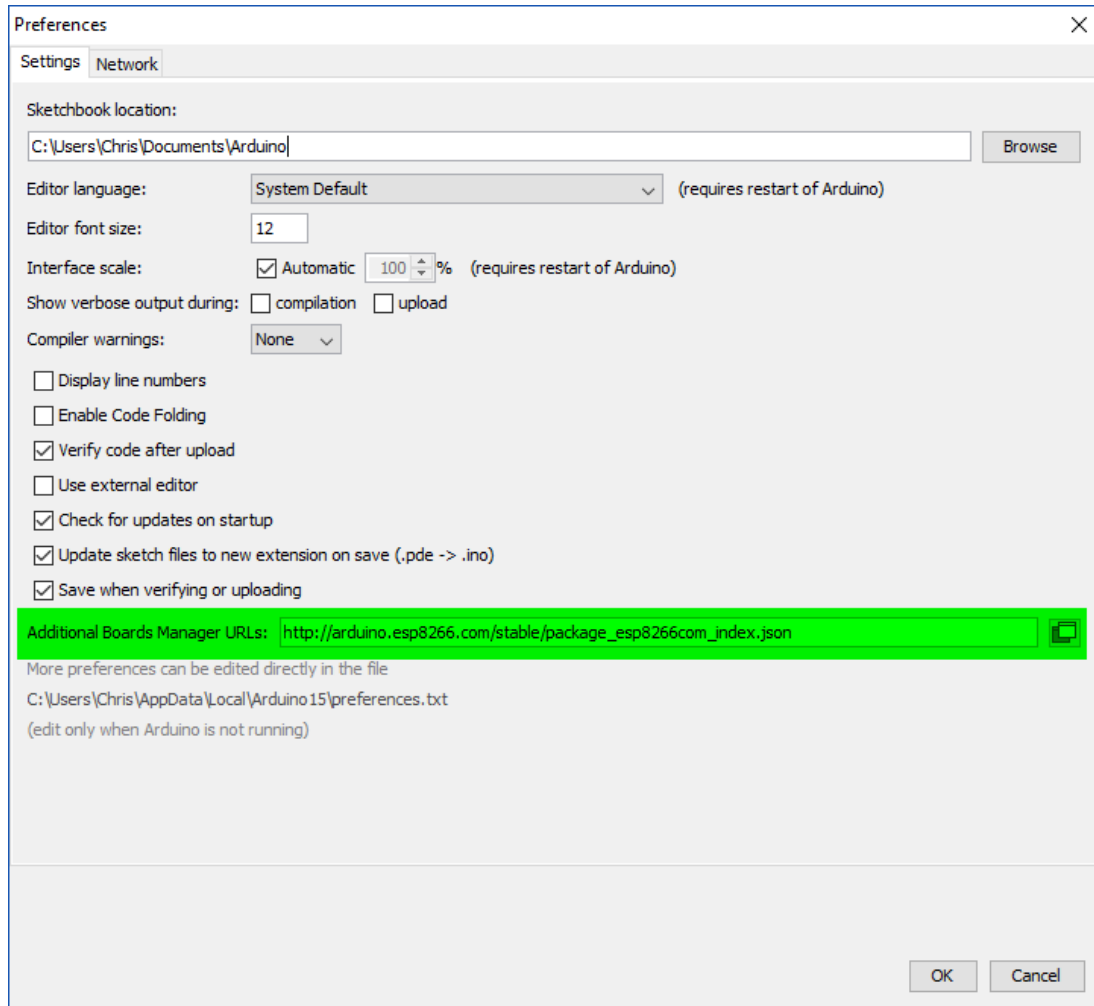
Download and setup the Arduino software referenced in '*Software Requirements*' section.

- Install the Arduino software and take all defaults.
- Launch the Arduino software
- Click on **File** → **Preferences**

Ensure that the new location for the ESP8266 board URL is configured as below:

Should be set to:

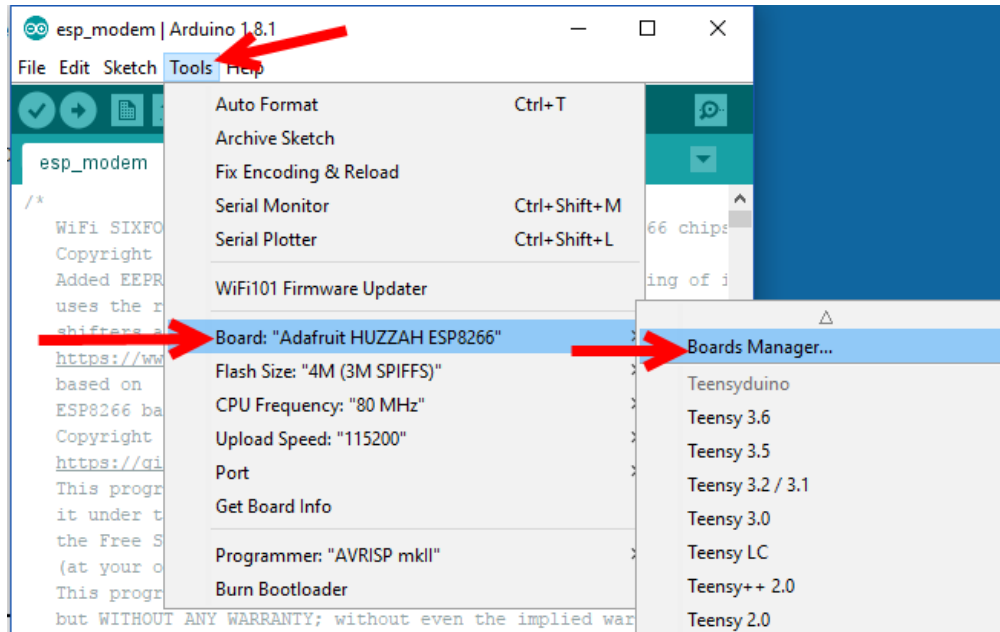
http://arduino.esp8266.com/stable/package_esp8266com_index.json



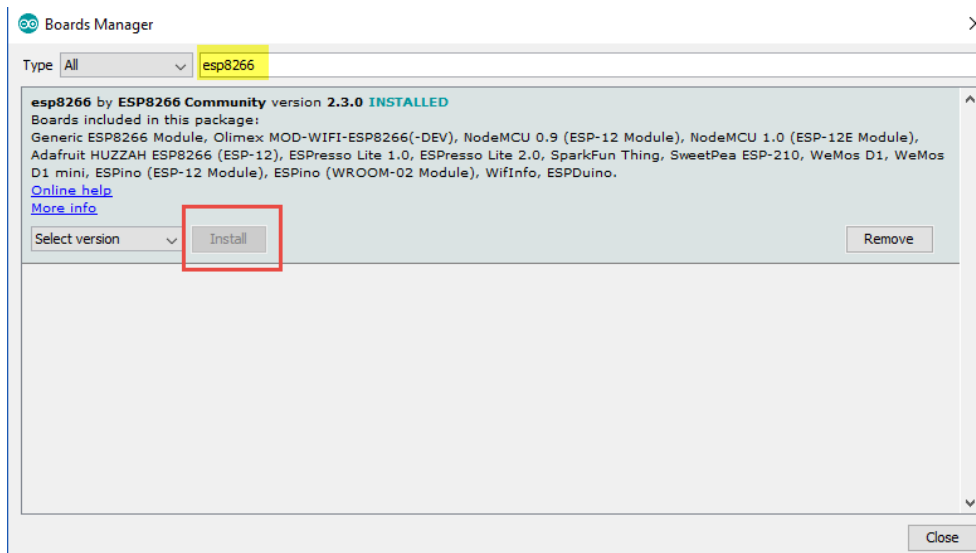
- Click OK

Verify the ESP8266 board is installed and up to date within the Board Manager area.

To Install/Update the Board Manager go to **Tools** → **Board: “....”** (May have text between the quotes) → **Boards Manager**

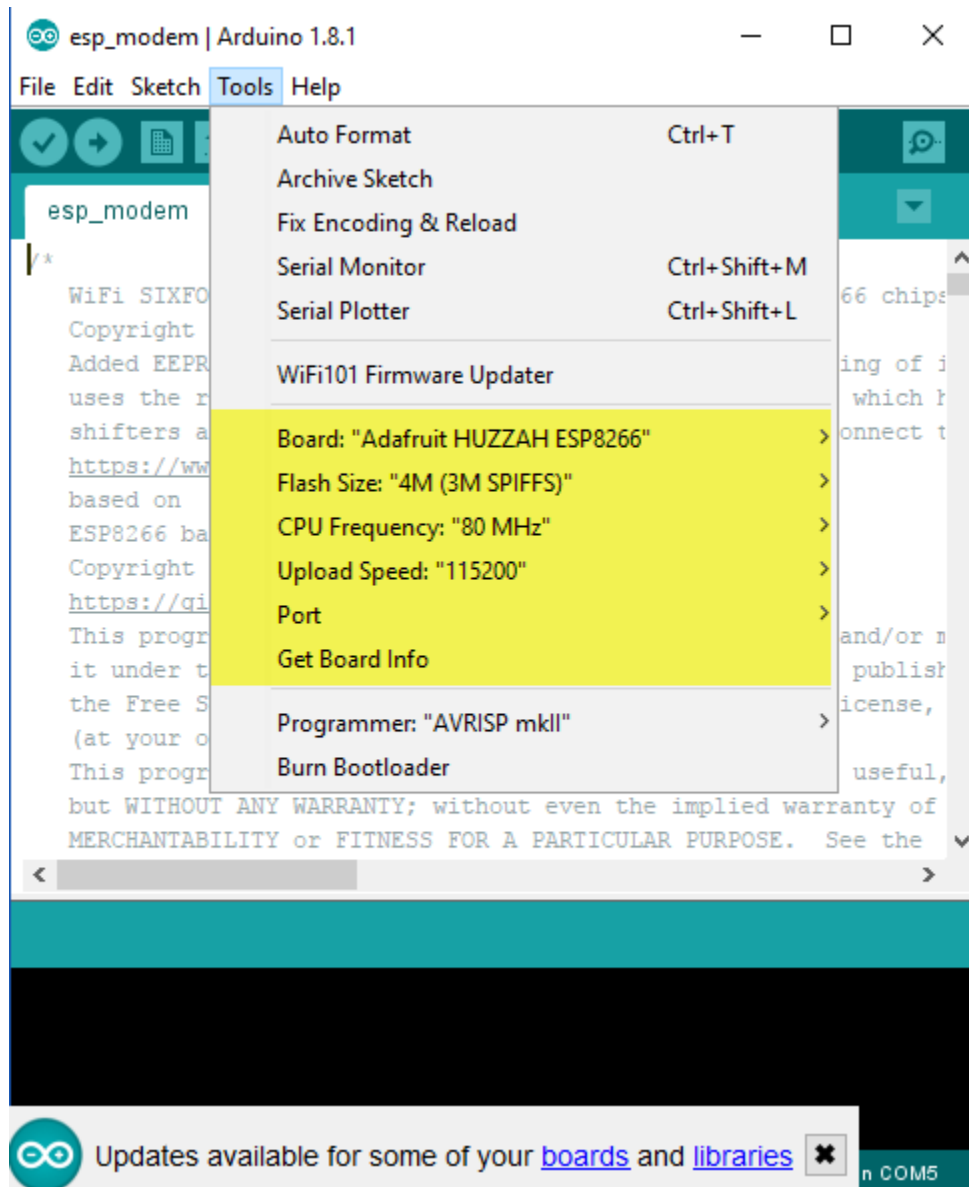


When the board manager comes up it will take a few seconds to find all the different boards that are loaded within the system. From there you can search for 'esp8266' (without quotes).

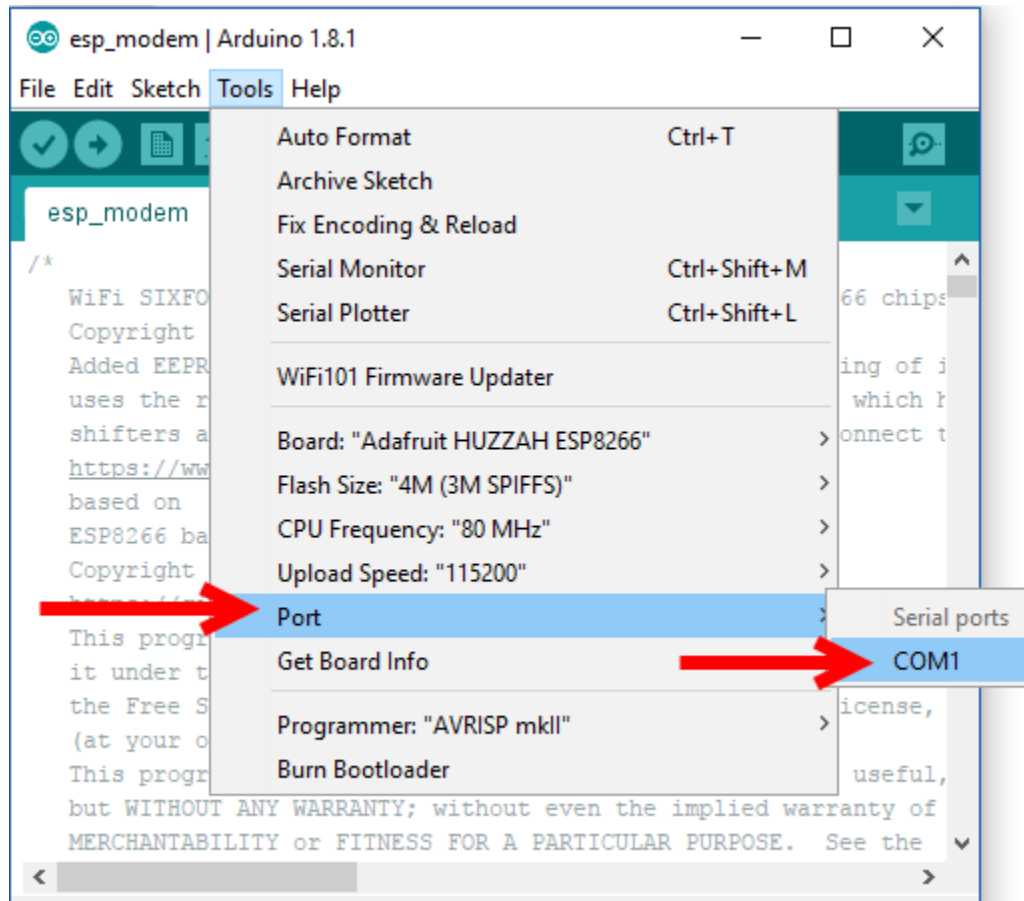


If it is not installed or needs updated the 'Install' button will be enabled.

- Click 'Install' to install/update the ESP8266 board modules.
- Click the 'Tools' menu and ensure the settings match as below:



- Ensure you select the correct Port that your FTDI was detected on your PC



9. Programming HUZZAH ESP8266 WiFi Board

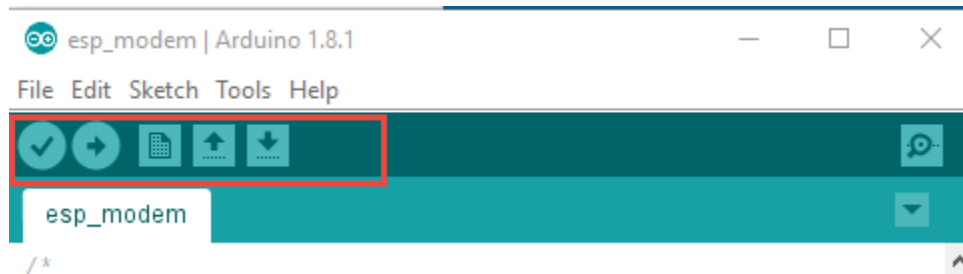
To program the HUZZAH ESP8266 you must use the FTDI cable (at least that's how I did it).


If the FTDI cable has not been connected Reference the '*Connecting the FTDI Programming Cable*' section to connect the FTDI cable to HUZZAH ESP8266 module.

- Verify what serial port has been assigned to the FTDI cable
- Launch the Arduino IDE software (if not already launched)
- Load the ESP8266 modem sketch downloaded from '*The Code*' section by selecting **File → Open**
- Navigate to the saved location of the Arduino sketch and open

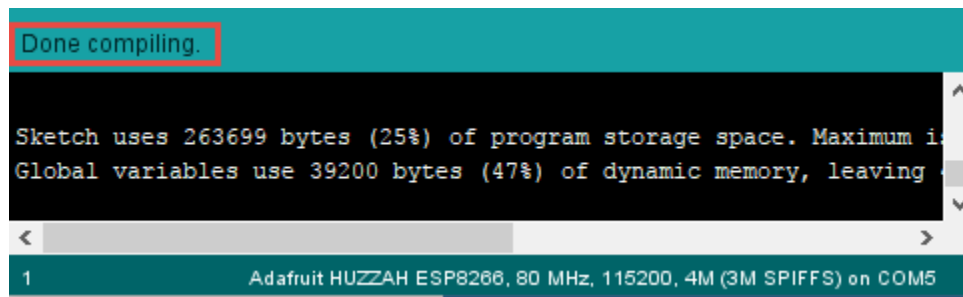
- The first time it opens the program will alert you it must be copied to a folder. Just click OK and it will do it for you. Usually located under My Documents\Arduino folder
- Verify board settings as outlined in ‘*Setting up Software for Use with HUZZAH ESP8266 WiFi Board*’ section.

Within your Arduino program there will be a quick launch bar of buttons:



- Click the  shown below to verify/compile the code to ensure there is no errors.

Within the status screen it should look like something similar as below:



If there are any errors it will be reported in this window.

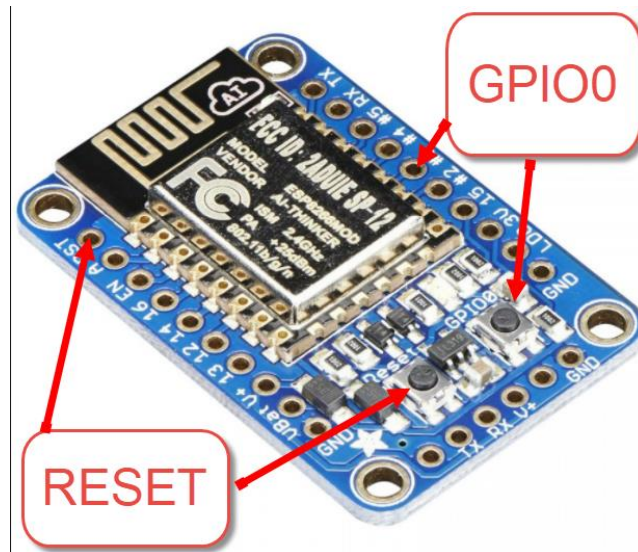
Once the compile is verified you can now program the board.


In order to program the ESP8266 you will now need to initiate the board into programming mode.

To initiate the HUZZAH ESP8266 into programming mode you will need to:

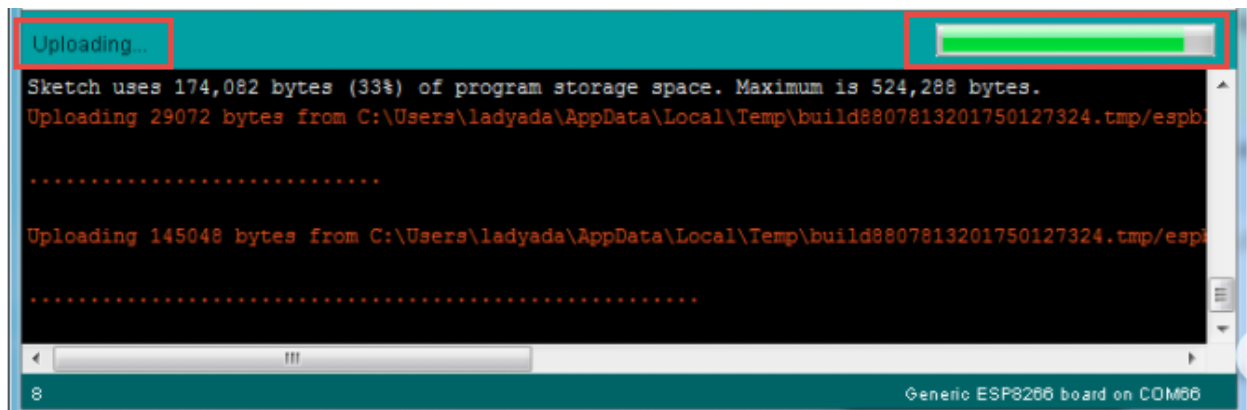
1. Hold down the **GPIO0** button, the red LED will be lit
2. While holding down **GPIO0**, click the **RESET** button
3. Release **RESET**, then release **GPIO0**

4. When you release the RESET button, the red LED will be lit dimly, this means its ready to program



- Click the  on the quick launch Arduino bar to start the programming of the board.

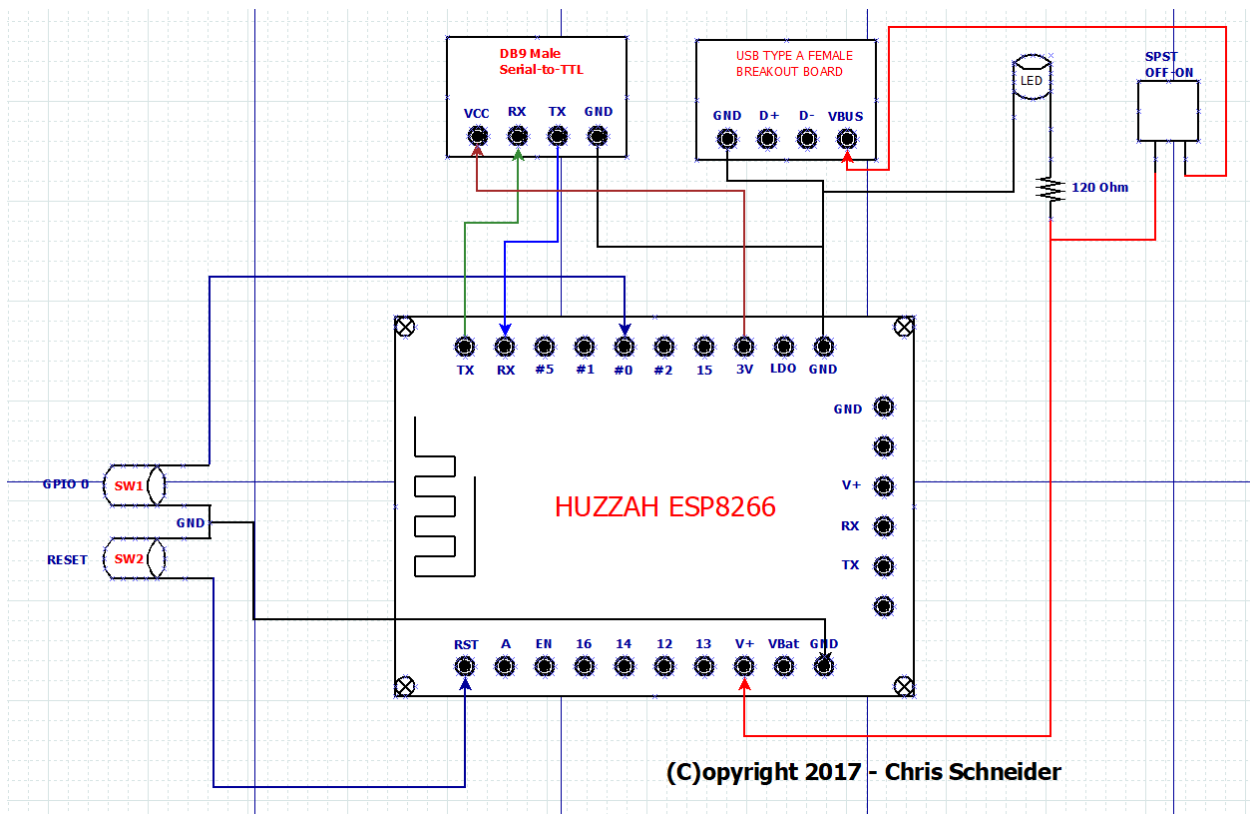
Once clicked it will recompile and program. The status windows should look similar to the below.

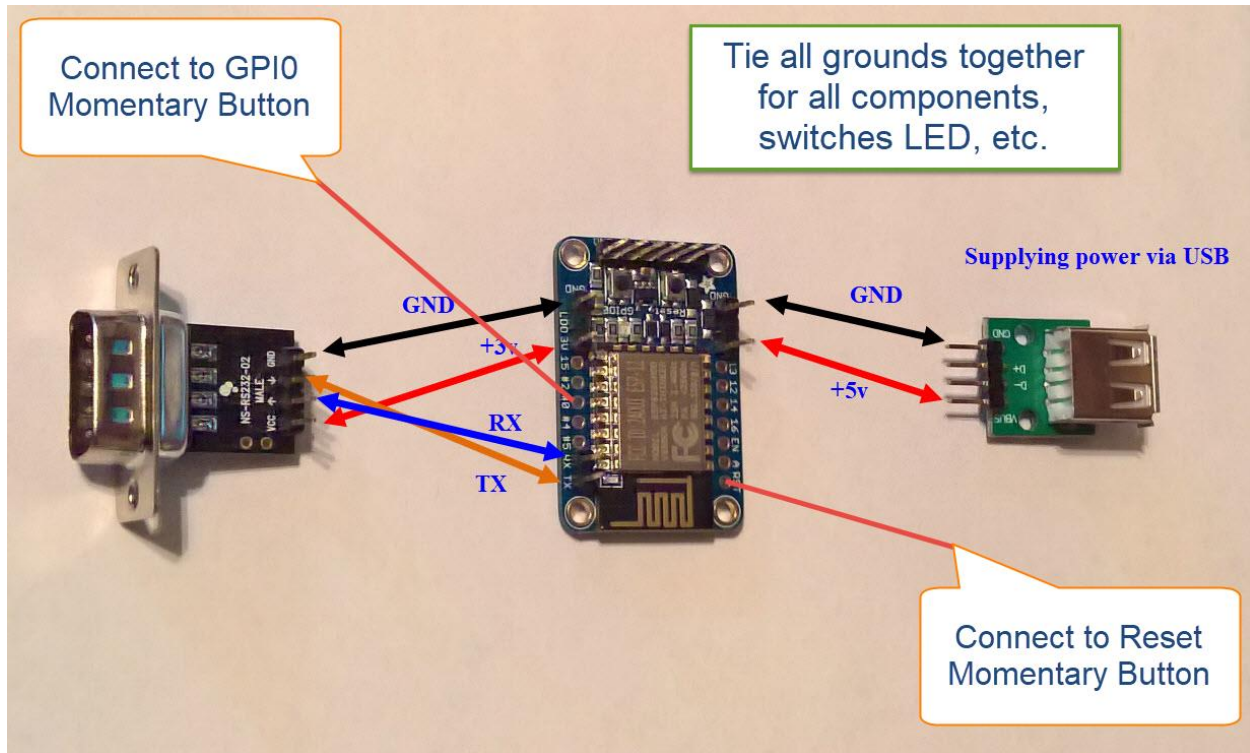


It will take a couple of minutes or so to upload the program. If there are any errors it will be reported in this window.

10.1 TI-WiFi Component / Cable Wiring

Schematic:





10.2 Cable Wiring Pinout

DSUB 9 Female PIN #	DSUB 25 Male PIN #
3	2
2	3
7	4
8	5
6	6
5	7
1	8
4	20

A Standard null modem cable will work if the unit will be hooked up to a standard PC serial port or NanoPEB.

11. Powering up the Unit

11.1 Connect Cables

Connect the required cables for power and serial communications.

11.2 First Time Boot

My experience is it's real easy to setup from a PC then move it to your retro computer.

The first time the unit is powered up after programming you will need to set your terminal to 300 baud. Power the unit up after the terminal program has been launched. Usually a garbage character(s) may appear. I typically after programming the unit and powering it up if there is no response when trying to type a command I will hold the GPIO0 down for at least 5 seconds and release then it will boot up. I have seen on occasion that you will have to press enter before it starts to boot up and you see the initial text display that includes firmware version, etc.

Once in then go ahead and set your baud rate (see Command Reference Section). As soon as the baudrate change command is issued the device will change the baud rate. The terminal will now need to be set to the new baud rate before you can save it.

Once back into the device with the new baudrate issue the AT&W command to write it to the NVRAM settings. This is now your permanent baudrate when the unit is powered up.

11.3 Ready to Go!

Now that you have the baud rate set and are ready. The below command should get you going:

AT+SSID=PUT IN YOUR WIRELESS SSID HERE <PRESS ENTER>

AT+PASS=PUT IN YOUR WIRELESS PASSWORD HERE <PRESS ENTER>

ATC1 <PRESS ENTER>

Now you should see the modem attempt to connect to the wireless. If it connects you will see:

Note: Your SSID/Network Name and IP's will be different.

CONNECTING TO SSID SKYNET.....

CONNECTED TO SKYNET

IP ADDRESS: 192.168.1.189

OK

If it fails to connect just issue the ATC1 command again.

To see all of your network information once it has connected:

ATI (not a L or a 1 but an upper case i)

You should see the below screen but with all your network information:

WIFI STATUS: CONNECTED

SSID.....: YOURSSID

MAC ADDRESS: MAC ADDRESS HERE

IP ADDRESS.: IPADDRESS

GATEWAY.....: GATEWAY IP

SUBNET MASK: NETMASK

SERVER PORT: 6400

WEB CONFIG.: HTTP://IPADDRESS

CALL STATUS: NOT CONNECTED

Now let's save the configuration so you don't have to do this again!

AT&W <Press Enter>

Now try to call out:

ATDT FUSIONBBS.DDNS.NET:9640

12. Command Set Summary

12.1 Help

An online help screen that shows the command set can be listed within the terminal by typing either of the below two commands:

AT?	ATHELP
-----	--------

12.2 Auto Answer

Enable/Disable Auto Answer incoming connection request. Issue the below command:

ATS0=N

N =	
0	Disabled
1	Enabled

12.3 Command Mode

To enter command mode after a connection has been established to a remote computer enter the below command within 1 second:

+++

12.4 Dial Host

To dial a host from the device once connected to the WiFi network simply initiating:

ATDTHOST:PORT#

12.5 Echo

Enable/disable local echo (Duplex) by issuing the below command:

N =	
0	Disabled

ATEN

1	Enabled
---	---------

12.6 Exit Command Mode

To exit command mode enter the below command:

ATO

12.7 Factory Defaults

To restore the device back to factory defaults issue the below command:

AT&F

12.8 Flow Control

Flow control can be enabled and disabled with the below command:

AT&KN (Where N=0 for None; N=1 for Hardware; N=2 for Software)

N =	
0	None
1	Hardware
2	Software

12.9 Hangup/Disconnect

To disconnect from a connected session enter the below command (must be in command mode):

ATH

12.10 HTTP GET

Fetch URL from internet. Will close connection once page is finished fetching.

ATGET<URL>

12.11 Load NVRAM

If changes were made and you want to reset them and have not saved current settings to NVRAM the below command can be used to reset back to the powered on default of the NVRAM.

ATZ

12.12 Network Info

Displays current network information such as IP address, WIFI Status, Gateway, Subnet Mask, etc. Issue below command:

ATI

12.13 PETSCII

Enable/disable PETSCII graphics. Terminal must support PETSCII if enabled. Issue below command to enable or disable:

ATPETN

N =	
0	Disabled
1	Enabled

12.14 Pin Polarity

To change the pin polarity issue the below command:

AT&PN (Where N=0/Inverse, 1/Normal) – Typically it's set to Normal and used by Commodore 64 computers only.

12.15 Query Commands

To query most command values above simply enter the command with a ? at the end. Do not enter the N value with the ? command. For example to retrieve the current baud rate enter:

AT\$SB?

12.16 Save to NVRAM

To save settings to NVRAM so that the settings stay resident when powered off and will be loaded upon next power up issue the below command:

AT&W

12.17 Set Baud Rate

To set the baud rate the WiFi Modem will use enter the below command:

AT\$SB=N

N for Baud Rates		
300	4800	38400
1200	9600	57600
2400	19200	115200

Note that currently the TI-99/4A and Myarc Geneve 9640 computers with TIMXT or Port (for Geneve) can max out at 38400 bps.

For Ti-99/4A's to use higher than 4800bps accurately the F18A must be installed in order to use the TIMXT terminal program.

12.18 Set Busy Message

To set a busy message to be displayed when a connection attempt is made (Auto Answer must be enabled to work) issue the below command:

AT\$BM=YOUR BUSY MESSAGE HERE

12.19 Set Speed Dial Entries

To set a host in order to use the speed dial command enter the below command:

`AT&ZN=HOSTNAME:PORT` (Where N=0 to 9)

Once created you will want to save it to NVRAM with the **AT&W** command.

12.20 Set Server Listening Port

The WiFi Modem firmware has the ability to answer incoming connection request on a specific port. By default the port is coded to 6400, but this can be changed. Issue the below command to change the port. To ensure the port is stored when the unit gets powered off the AT&W command will also have to be issued to save it.

`AT$SP=N` (Where N = 1 to 65535)

12.21 Set WiFi Password

Before connecting to a WiFi network a WiFi Password must be set. To set the WiFi Password enter the below command:

`AT$PASS=YOUR WIFIPASSWORD HERE`

12.22 Set WiFi SSID

Before connecting to a WiFi network a WiFi SSID must be set. To set the SSID (network name given to the WiFi) enter the below command:

`AT$SSID=YOUR WIFISSID HERE`

12.23 Show Settings

To show current active settings and NVRAM settings issue the below command:

`AT&V`

12.24 Speed Dialing Stored Entries

Speed dialing host can be done if the host is stored into the NVRAM settings. Host are stored in slots 0–9. To speed dial simply initiate the below command:

ATDSN (Where N=0 to 9)

12.25 Telnet Command Handling

Enables/disables telnet command handling. Typically set to disabled.

ATNETN

N =	
0	Disabled
1	Enabled

12.26 Verbose Mode

To enable/disable verbose response codes enter the below command:

ATVN

N =	
0	Disabled
1	Enabled

12.27 WiFi (Connect/Disconnect)

To connect or disconnect to and from the WiFi network after the WiFi SSID and WiFi Password have been set issue the below command:

ATCN

N =	
0	Disconnect From WiFi
1	Connect To WiFi

13. TI/Geneve BBS (via Telnet)

The below BBS' can be connected to over the internet with this device or any other telnet based device/pc.

BBS Name	Address	Port
FuSiON BBS	fusionbbs.ddns.net	9640
Heatwave	heatwave.ddns.net	9640
The Keep	www.thekeep.net	23

14. Support

If anyone needs help with this project feel free to reach out to me and I will do the best that I can.

Shift838@att.net

<http://shift838.99er.net>