**Setting up ESP8266 and Using it**

**Author: Rishi Thakkar**

**Version: 01**

**6/13/2015**

**Contents**

[1. Wiring the ESP8266 3](#_Toc422482080)

[1.1 Normal Mode: Wiring 3](#_Toc422482081)

[1.2 Normal Mode: AT-Commands 3](#_Toc422482082)

[1.3 Flash mode 4](#_Toc422482083)

[2. Flashing the ESP8266 with NodeMCU 4](#_Toc422482084)

[2.1 Flashing Steps 4](#_Toc422482085)

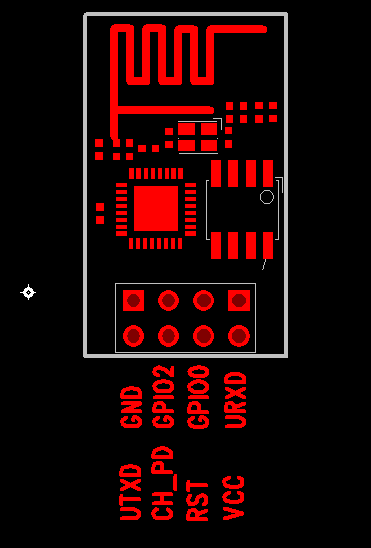
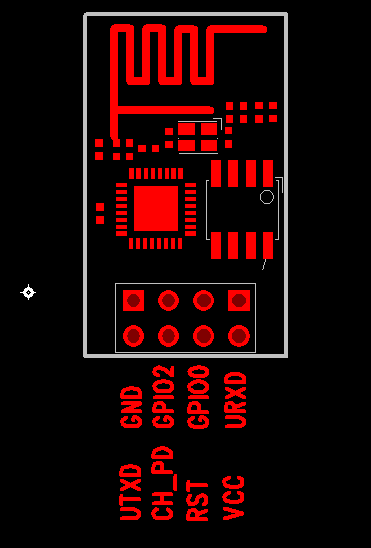
[2.2 LUA Uploader 6](#_Toc422482086)

[2.3 ESPlorer 6](#_Toc422482087)

[3. Helpful Websites 6](#_Toc422482088)

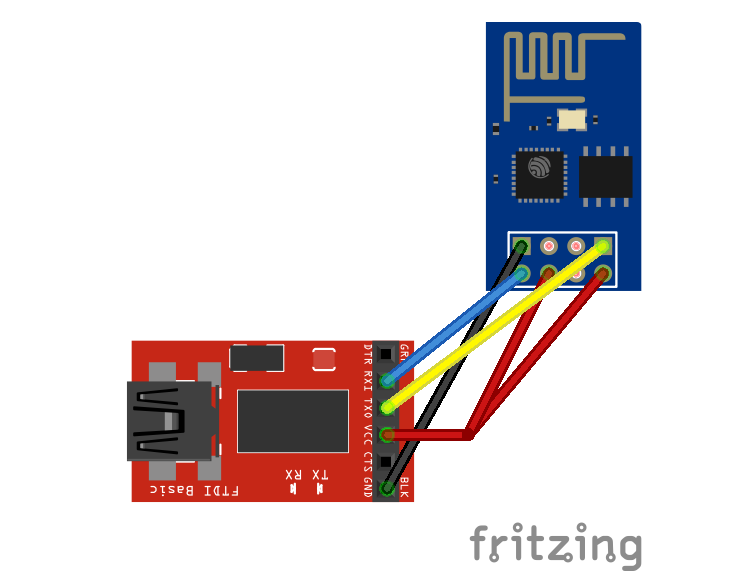
# Wiring the ESP8266

\*In both of these cases, place a pull up 10k resistor between Vcc and CH\_PD.

1. Normal Mode: Wiring

The wiring set up for the normal working mode is shown below in the image.



* VCC needs 3.3V
* CH\_PD has to be pulled-up (meaning it has to be connected to 3.3V as well)
* GND is connected to FTDI's GND pin
* RX is connected to FTDI's TX pin, because you want to create a loop: RX-> TX => RX -> TX
* TX is connected to FTDI's RX pin
* other pins are left floating

Depending on the firmware version, the baud rate is different: 9600, 57600 or 115200. Try all of these before you think that there is a problem.

To see if it's working, type in AT, if the module replies with OK then you are in the green.

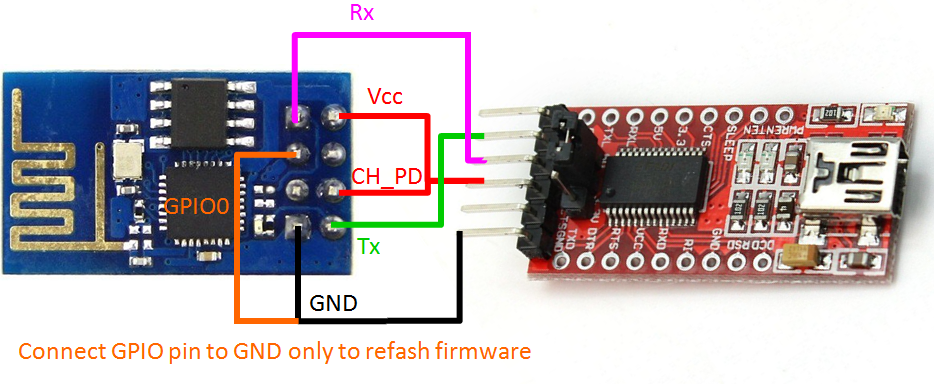
1. Normal Mode: AT-Commands

The whole command set is available on github (https://github.com/espressif/esp8266\_at/wiki/AT\_Description). The four basic types are shown below:

| **Type** | **Instruction Format** | **Description** |
| --- | --- | --- |
| Test | AT+CMD=? | Query the Set command or internal parameters and its range values. |
| Query | AT+CMD? | Returns the current value of the parameter. |
| Set | AT+CMD=Parameter | Set the value of user-defined parameters in commands and run. |
| Execute | AT+CMD | Runs commands with no user-defined parameters. |

1. Flash mode

For the flash more the wiring is a little different. This is shown in the picture below.



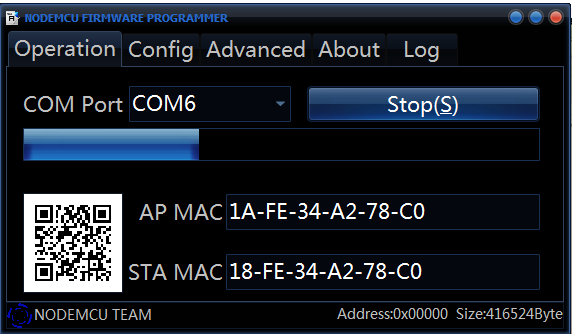
* VCC needs 3.3V
* CH\_PD has to be pulled-up (meaning it has to be connected to 3.3V as well)
* GND is connected to FTDI's GND pin
* RX is connected to FTDI's TX pin, because you want to create a loop: RX-> TX => RX -> TX
* TX is connected to FTDI's RX pin
* GIPO 0 is connected to ground. This is the difference.
* other pins are left floating

The instructions for flashing will be provided in the next section.

1. Interface/External Fixture

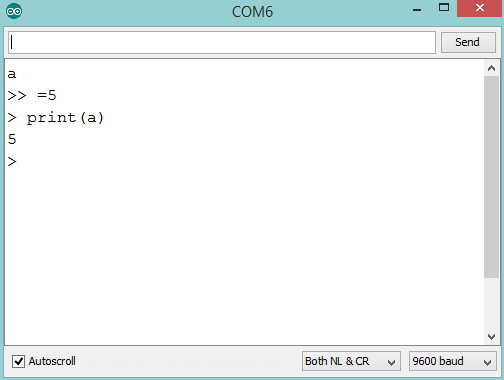
|  |  |
| --- | --- |
| /Users/rishithakkar/Documents/Projects/_ECEProjects/ECEProjects/Home Automation Projects/_ESP8266-Getting started/refrenceForInterface_ESP8266.png | Red box: Input ports used for the ESP8266  Orange box: Input ports used when we want to be in flash mode  Green box: Input ports used when we want to be in regular programing mode.  Red wire: Vcc of 3.3 V (3rd hole from the A)  Black wire: Ground (1st hole from the A)  Blue wire: Tx on the FTDI board (4th hole from the A)  Green wire: Rx on the FTDI board (5th hole from the A) |

# Flashing the ESP8266 with NodeMCU

1. Flashing Steps
2. Need to download NodeMCU from github (https://github.com/nodemcu/)
3. Download the NodeMCU flasher and unzip it.
4. Then flash the chip:

\*As you can see this chip has 2 MAC addresses

1. Now remove the grounding for GPIO 0 and reboot the chip.
2. Switch the baud rate to 9600 and try out some commands.



You are now talking to the Lua interpreter. This is useful if you want to do a couple of things at a time but it soon becomes a pain when you have large amounts of code that you want to upload. This is where you use the Lua uploader or ESPlorer.

Go to https://github.com/nodemcu/nodemcu-firmware/wiki/nodemcu\_api\_en for the API instructions.

1. LUA Uploader
2. Download the lua uploader which is available on github (https://github.com/hwiguna/g33k/tree/master/ArduinoProjects/Windows/ESP8266\_Related)
3. You can run code immediately and even store code in terms of a file on the esp.
4. ESPlorer
5. Download the lua uploader which is available on github (https://github.com/hwiguna/g33k/tree/master/ArduinoProjects/Windows/ESP8266\_Related)

This does the same things as the LUA Uploader. They both have their pros and cons. It all depends.

# Helpful Websites

http://williamdurand.fr/2015/03/17/playing-with-a-esp8266-wifi-module/

http://fab.cba.mit.edu/classes/863.14/tutorials/Programming/serialwifi.html

http://vaasa.hacklab.fi/2015/02/17/esp8266-troubleshooting/

http://tomeko.net/other/ESP8266/

http://www.electrodragon.com/w/ESP8266#Setup\_Check\_list

https://importhack.wordpress.com/2014/11/22/how-to-use-ep8266-esp-01-as-a-sensor-web-client/

http://nodemcu.com/index\_en.html

<http://www.lua.org/>