

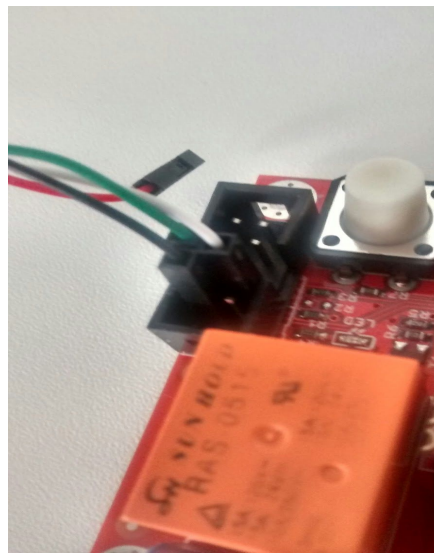
This is a guide for setting up an ESP8266 Lua-based programming environment on Windows.

[Lua](#) is a lightweight, high-level scripting language compiled in ANSI-C. It's commonly used in the video game industry, but has been adapted for use in ESP8266 programming because it's fast, portable, and small (the tarball for Lua 5.3.1, which contains source code and documentation, takes 276Kb compressed and 1.1Mb uncompressed).

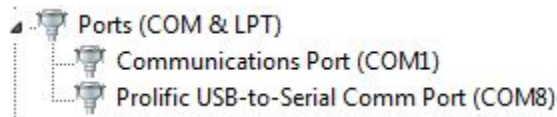
If you're looking to learn Lua, I recommend using [Nova Fusion's Lua For Programmers](#) to get the basics (more than enough to get going), and afterwards using [Programming in Lua](#) as a reference guide and for extra details on specifics of the language. Programming in Lua is the *K&R* of Lua; it was written by Lua's chief architect and remains the most widely used source for gaining a solid base in Lua. If you're impatient and just want to get straight into the code, I recommend [Learn X in Y Minutes: Lua](#).

There are two major ESP8266 lua-based firmwares: [NodeLua](#) and [NodeMCU](#). We're going to flash NodeMCU onto the ESP8266 purely because the development community is more active, and so troubleshooting is much easier in the long run.

First of all, connect your ESP8266-EVB to your PC using a USB-to-TTL serial cable like so: Black goes to ground, Green goes to Rx, White goes to Tx. Schematics of the board can be seen [here](#).

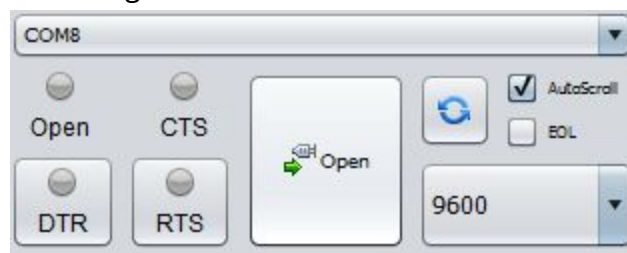


Turn on your ESP8266 by plugging in the power supply. The red LED should turn on to indicate the device has booted. If the connection is not recognised on your PC (you don't see *Prolific USB-to-Serial Comm Port* under the *Ports (COM & LPT)* section in the Device Manager), then you need to download and install the necessary Windows [drivers](#) for the cable.

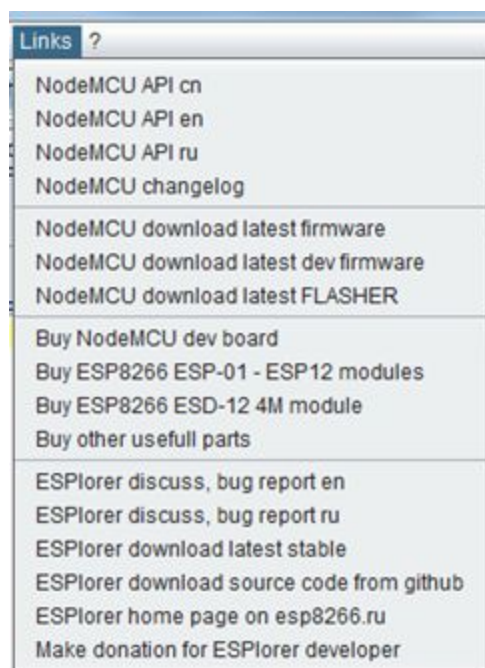


Next you should download [ESPlorer](#), an IDE for creating and downloading Lua programs to your ESP8266. Click the big blue dodgy-looking button on that web page to download the latest version. Unzip it, and run ESPlorer.jar (it's a jar file, so you'll need Java).

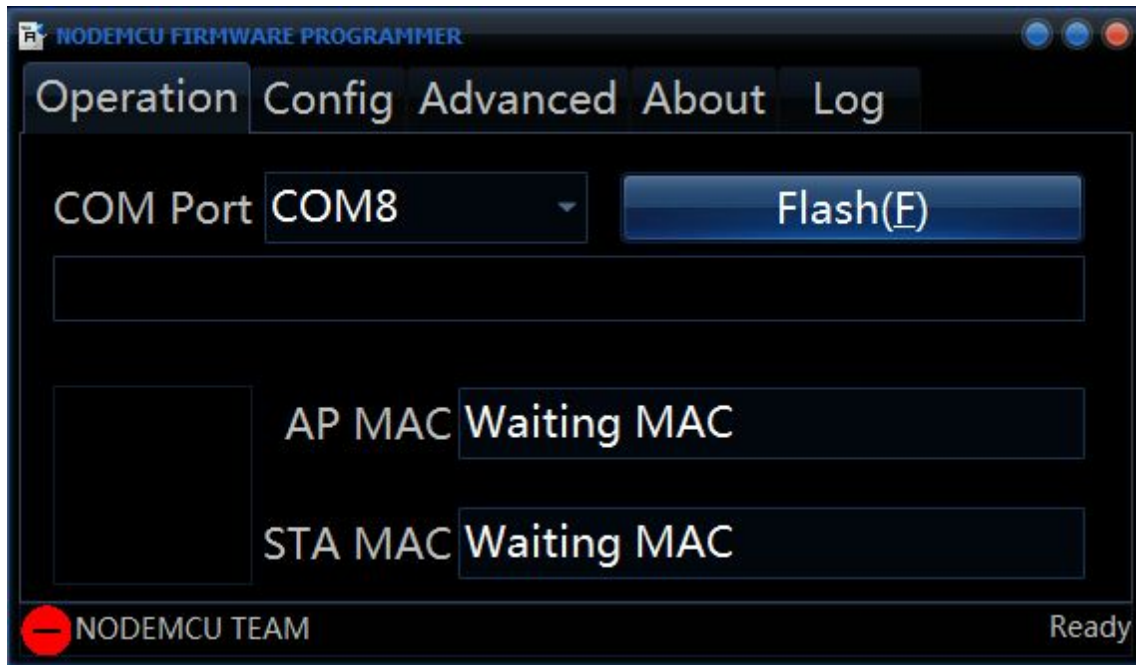
Connect to your ESP8266 by going to the top right part of the screen, choosing the correct COM port (find this out from Device Manager), and choose the correct baud rate (the rate at which information is transferred in the communication channel). It will most likely be 9600, but if you are having trouble communicating with your device it's worth trying out a few different levels. Click Open to begin communication. Your device is now connected and communicating with the PC.



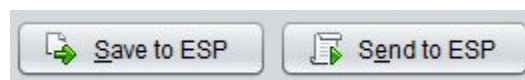
Now we're going to download and flash the latest stable version of NodeMCU to our device. On new versions of ESPlorer (v0.2.0 and onwards), we can use the IDE to easily download the firmware. From the Links menu at the top of the application, choose *NodeMCU Download Latest Firmware*. Save the bin file. Now choose from the Links menu *NodeMCU Download Latest FLASHER*. Click *Download ZIP* on the right-hand side of the opened Github page. Now close ESPlorer.



Unzip the flasher folder, and navigate to either the Win32 or Win64 folder, depending on your version of Windows. Run the flasher.exe in that folder. Go to the Config tab, select the gear next to *Internal://NodeMCU* (the green one), and select the nodemcu_latest.bin file you downloaded earlier. Go back to the Operations tab, choose the correct COM port (same as before), and press FLASH.



Once the flash completes (a green tick appears at the bottom), close the flasher, and reopen ESPlorer. Connect to the device as before. You can now begin writing Lua programs on the left-hand side of the application. Name your "main" file init.lua, and send or save it to the ESP using the buttons at the bottom of the screen.



Sources:

[Rui Santos' Random Nerd Tutorials](#)

[TornTech's ESP8266 Complete Guide](#)

NodeMCU Resources:

[Some examples](#)

[API](#)