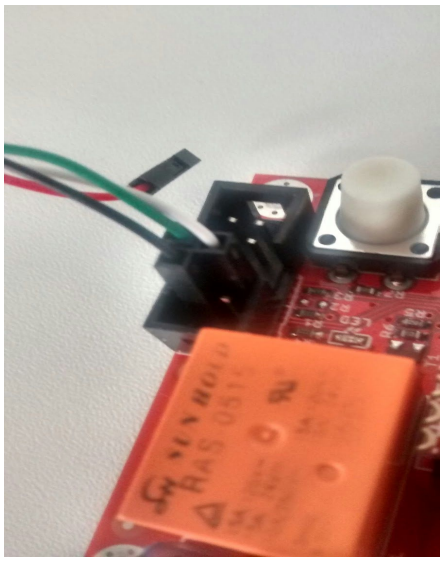
This is a guide for setting up an ESP8266 Lua-based programming environment on Linux.

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. On Linux you shouldn't require any extra drivers; to check it connected go to a terminal and enter the command `lsusb`, you should see something like:

```
joseph@joseph-Inspiron-7537:~$ lsusb
Bus 001 Device 002: ID 8087:8000 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 002 Device 005: ID 04f3:0206 Elan Microelectronics Corp.
Bus 002 Device 004: ID 8087:07dc Intel Corp.
Bus 002 Device 003: ID 0c45:6705 Microdia
Bus 002 Device 002: ID 067b:2303 Prolific Technology, Inc. PL2303 Serial Port
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
joseph@joseph-Inspiron-7537:~$
```

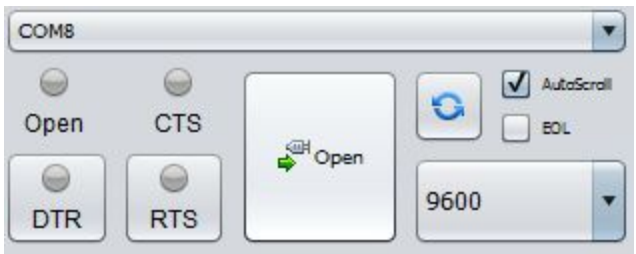
In my case the usb serial converter has been listed as “Prolific Technology, Inc. PL2303 Serial Port”. Then if you enter the command `dmesg | grep tty` you should see something like this:

```
joseph@joseph-Inspiron-7537:~$ dmesg | grep tty
[    0.000000] console [tty0] enabled
[    4.306090] usb 2-3: pl2303 converter now attached to ttyUSB0
joseph@joseph-Inspiron-7537:~$
```

This shows which USB port the converter is connected to. As you can see it’s connected to ttyUSB0.

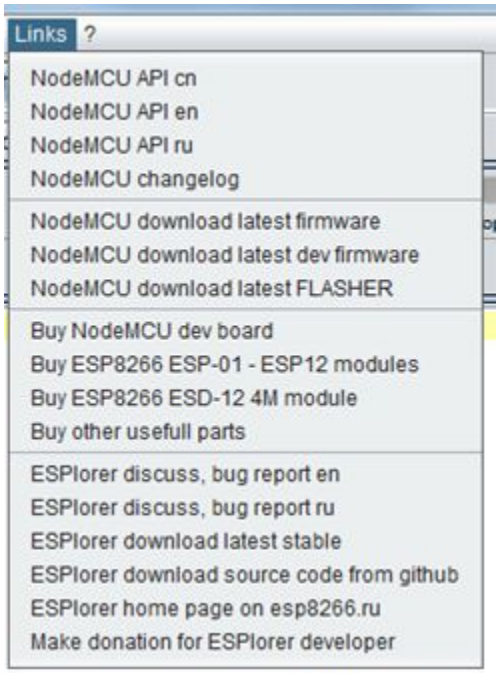
Next you should download ESPlorer, an IDE for creating and downloading Lua programs to your ESP8266. Click the big blue dodgy-looking button on that web page to download the latest version. Unzip it, and run ESPlorer.jar (it’s a jar file, so you’ll need Java).

Connect to your ESP8266 by going to the top right part of the screen, choosing the correct port (which you learned from above, it’ll be something like *ttyUSB0*), 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 being 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 close ESPlorer.



Now, download esptool by cloning the repository:

```
git clone https://github.com/themadinventor/esptool.git
```

Then, open a terminal and go to the folder you cloned esptool into. Use this command to flash the firmware you just downloaded onto the device:

```
sudo python esptool.py --port /dev/ttyUSB0 write_flash 0x00000 NodeMCU_FirmwarePath/NodeMCU_Firmware.bin
```

replacing ttyUSB0 with your port, and adding in your own path to the firmware bin file you downloaded.

Once the flash completes, 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
http://www.whatimade.today/flashing-the-nodemcu-firmware-on-the-esp8266-linux-guide/

NodeMCU Resources:
Some examples
API