

# Tutorial for programming ESP8266 on windows, using the NodeMCU devkit

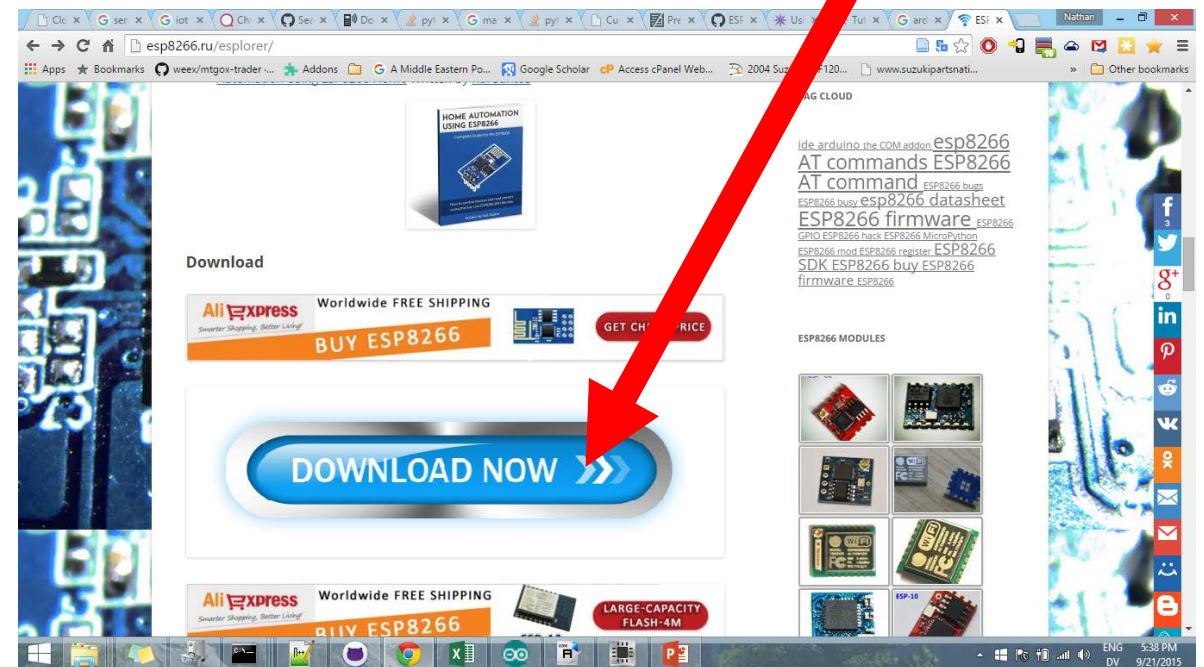
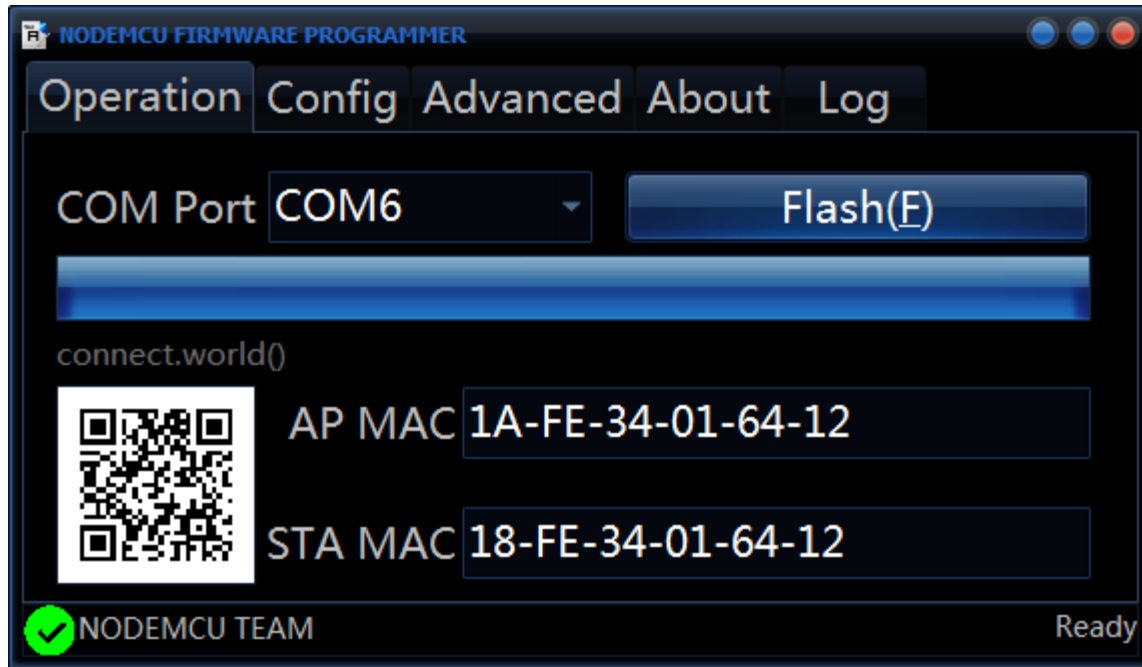
Get the nodemcu flasher from here:

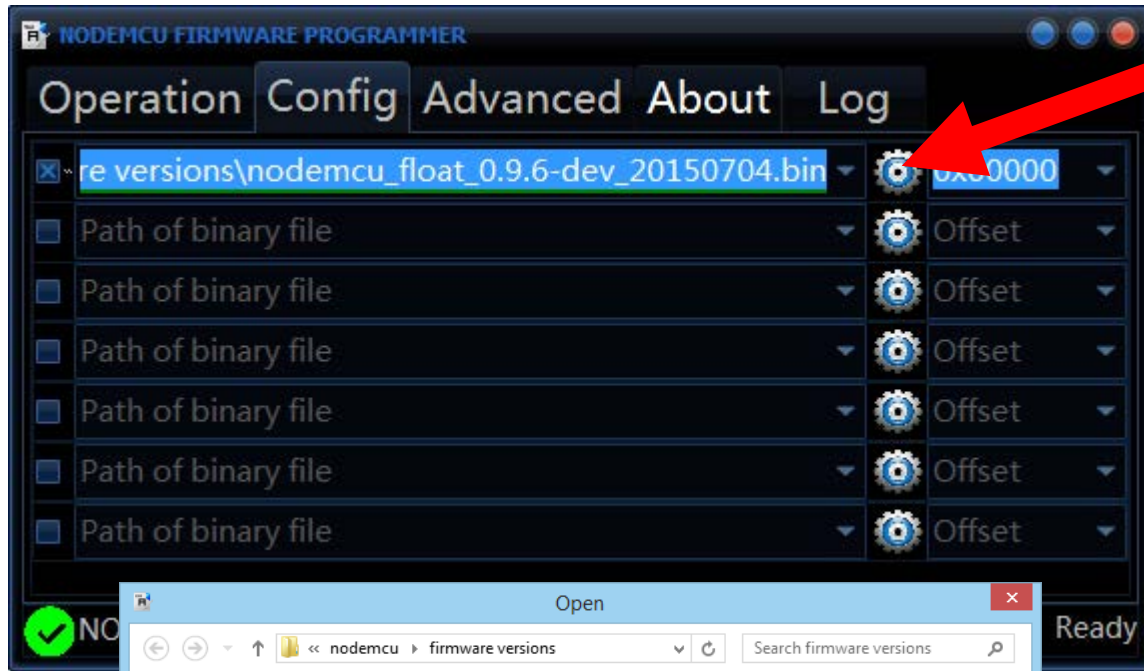
<https://github.com/nodemcu/nodemcu-flasher>

Get ESPlorer from here:

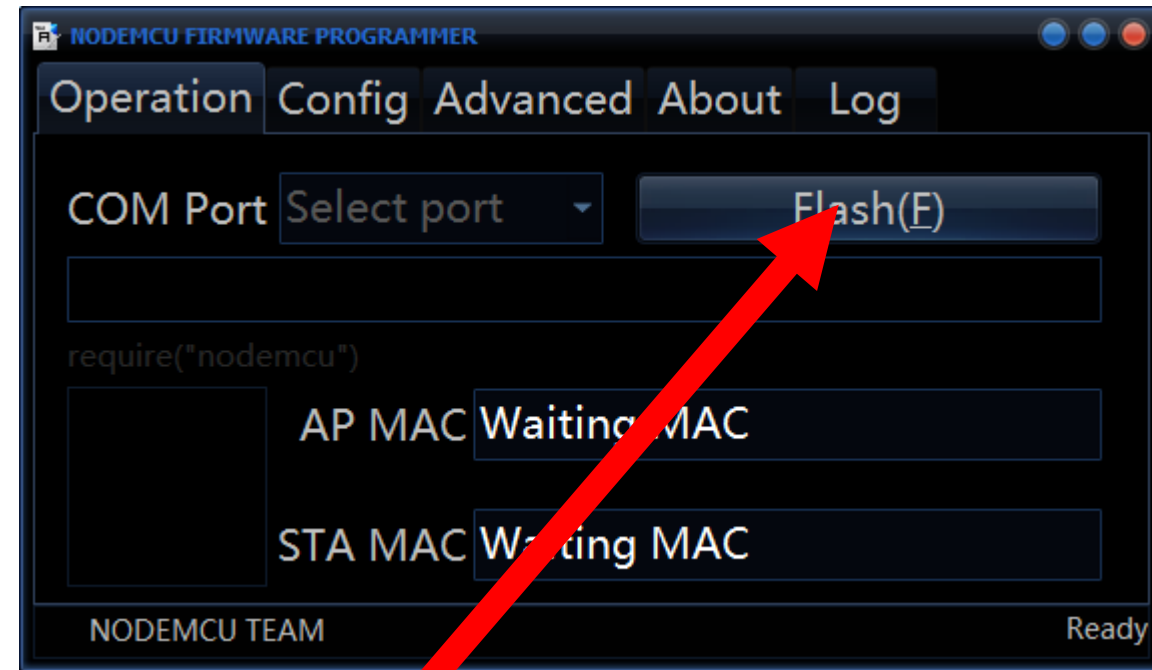
<http://esp8266.ru/esplorer/>

Or from the [github page](#), though I've always used the Russian Link.

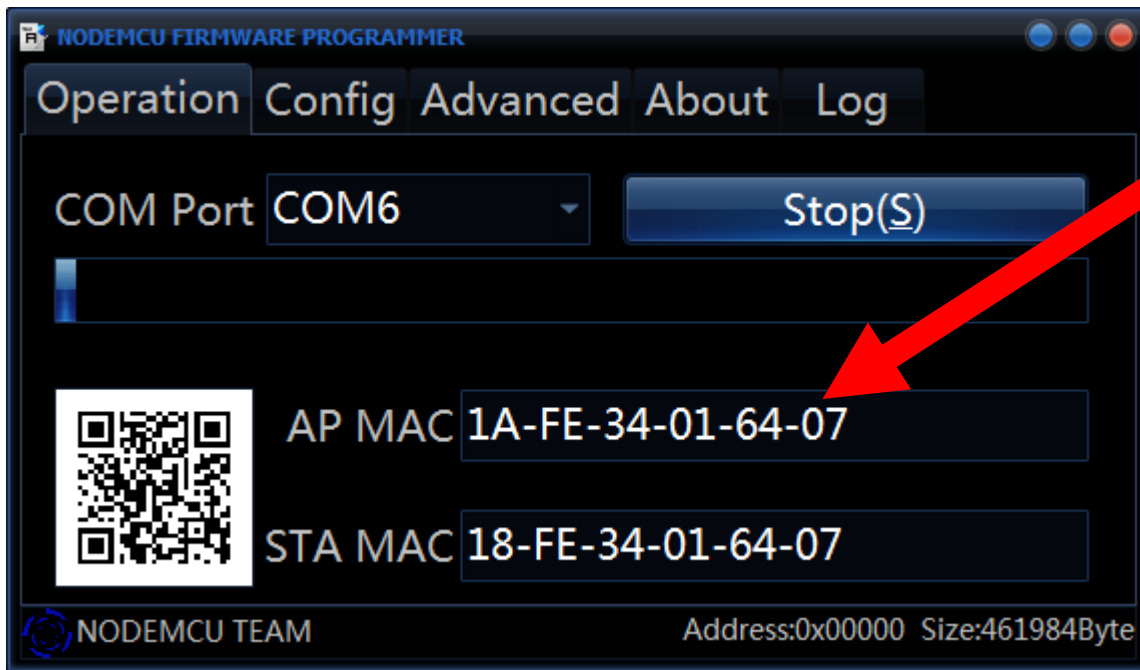




First, click on the gear under config, and choose your firmware file. Use the latest from <https://github.com/nodemcu/nodemcu-firmware/releases>, which you can find by googling 'nodemcu releases'

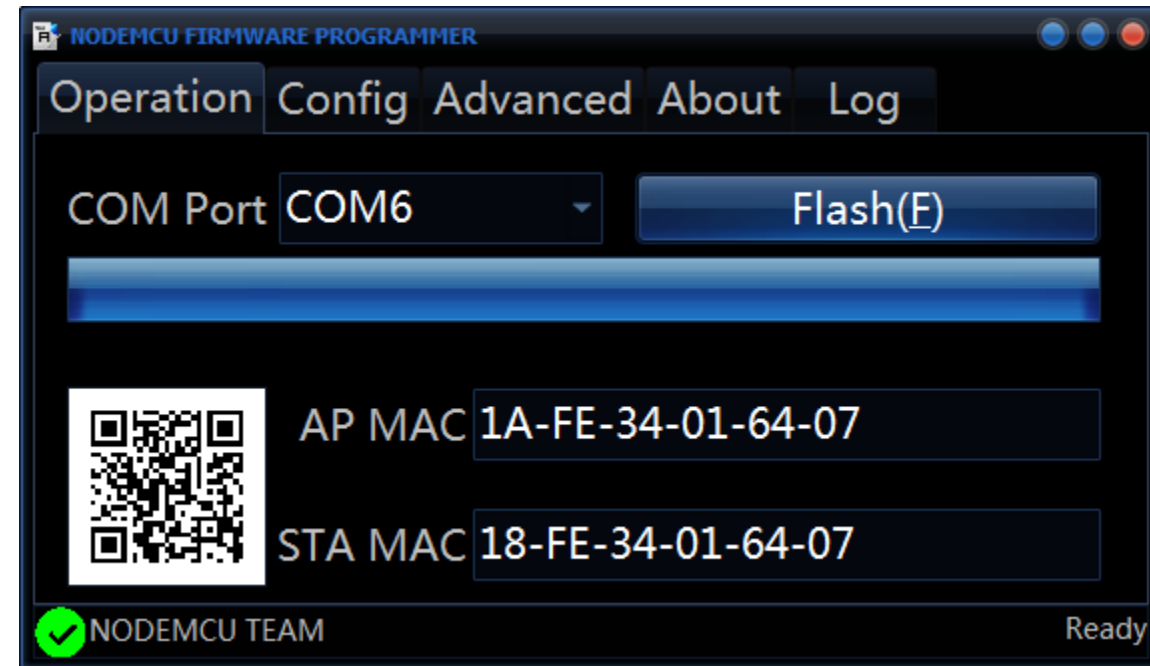
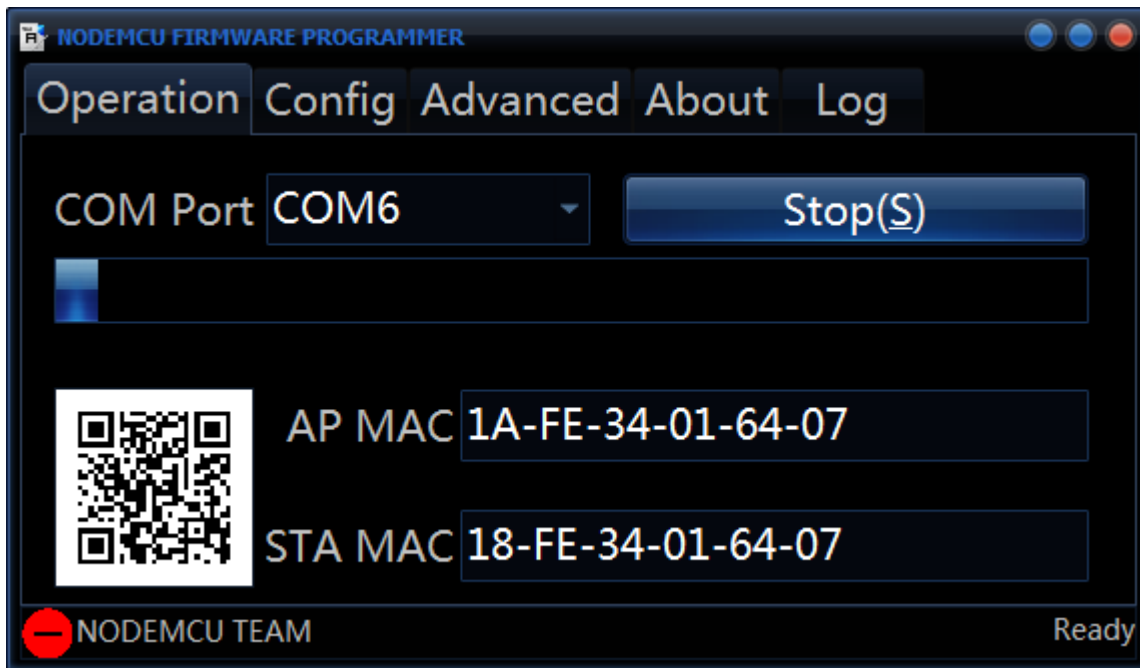


Next, click 'flash'

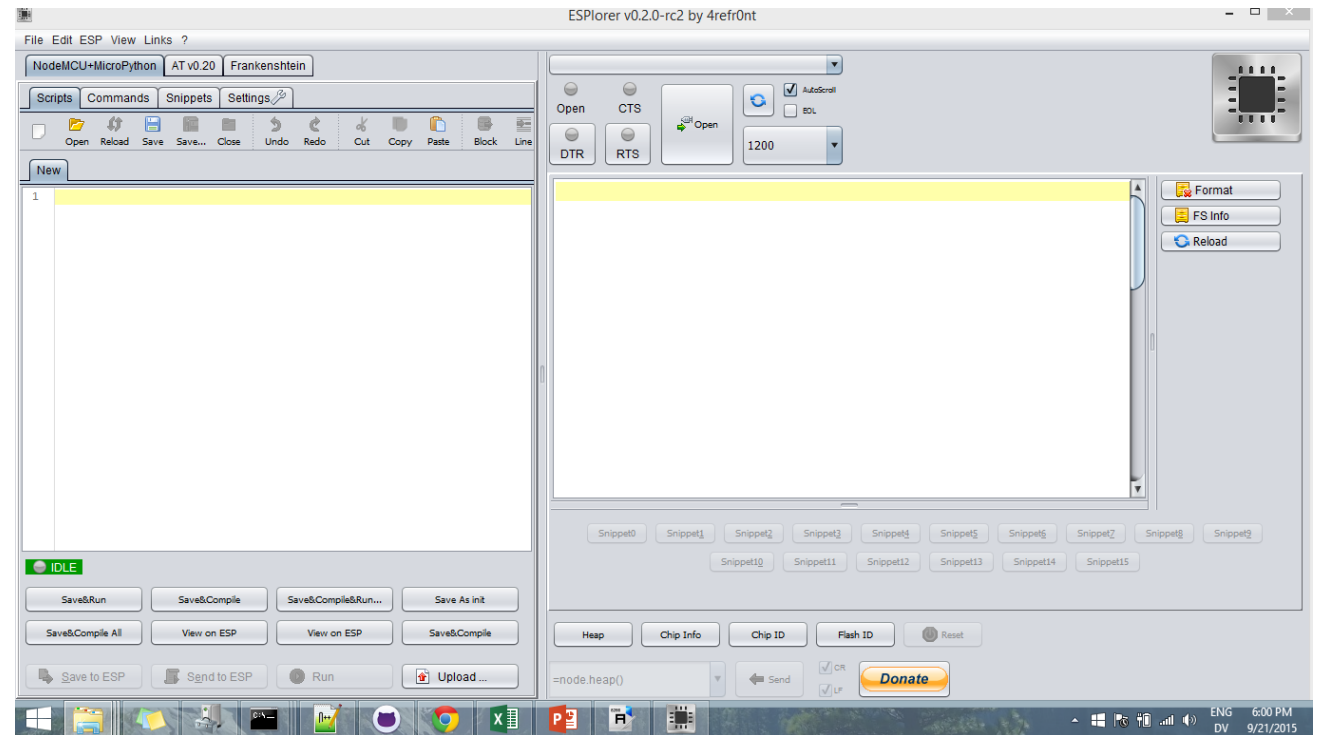
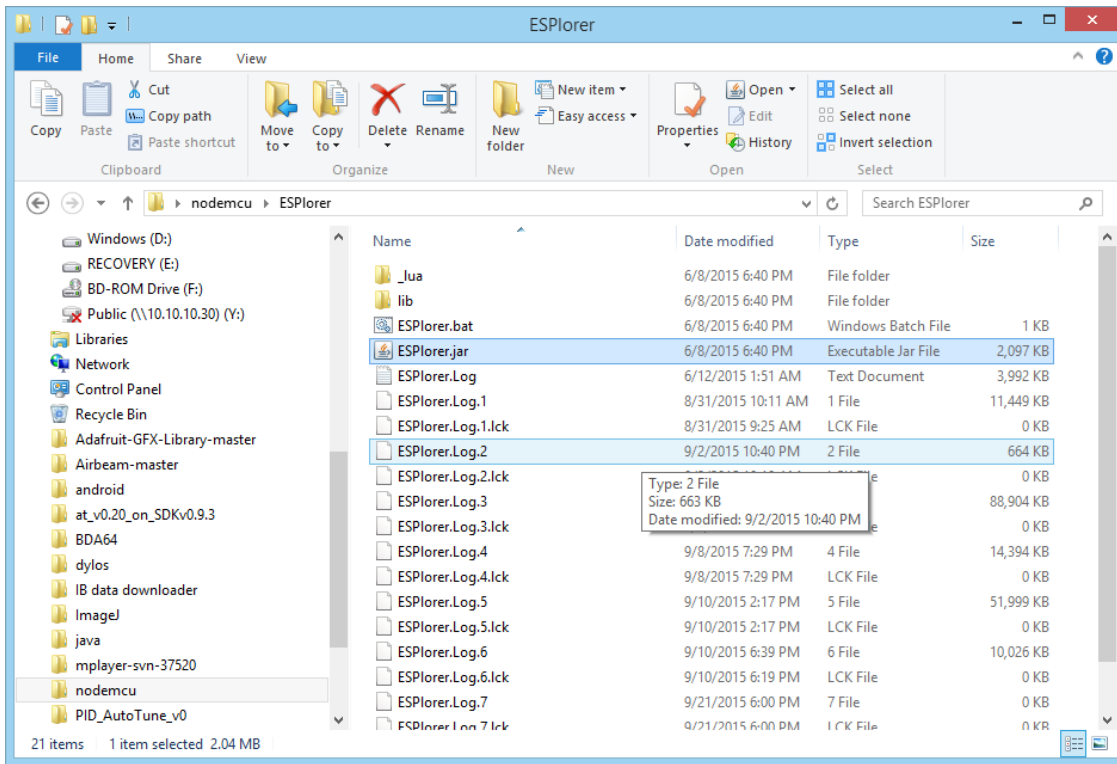


The MACs will be filled in along with the QR code. If something goes wrong, the red circle will appear in the bottom left, otherwise, once the status bar gets to full, and everything went alright, the status circle will change to green.

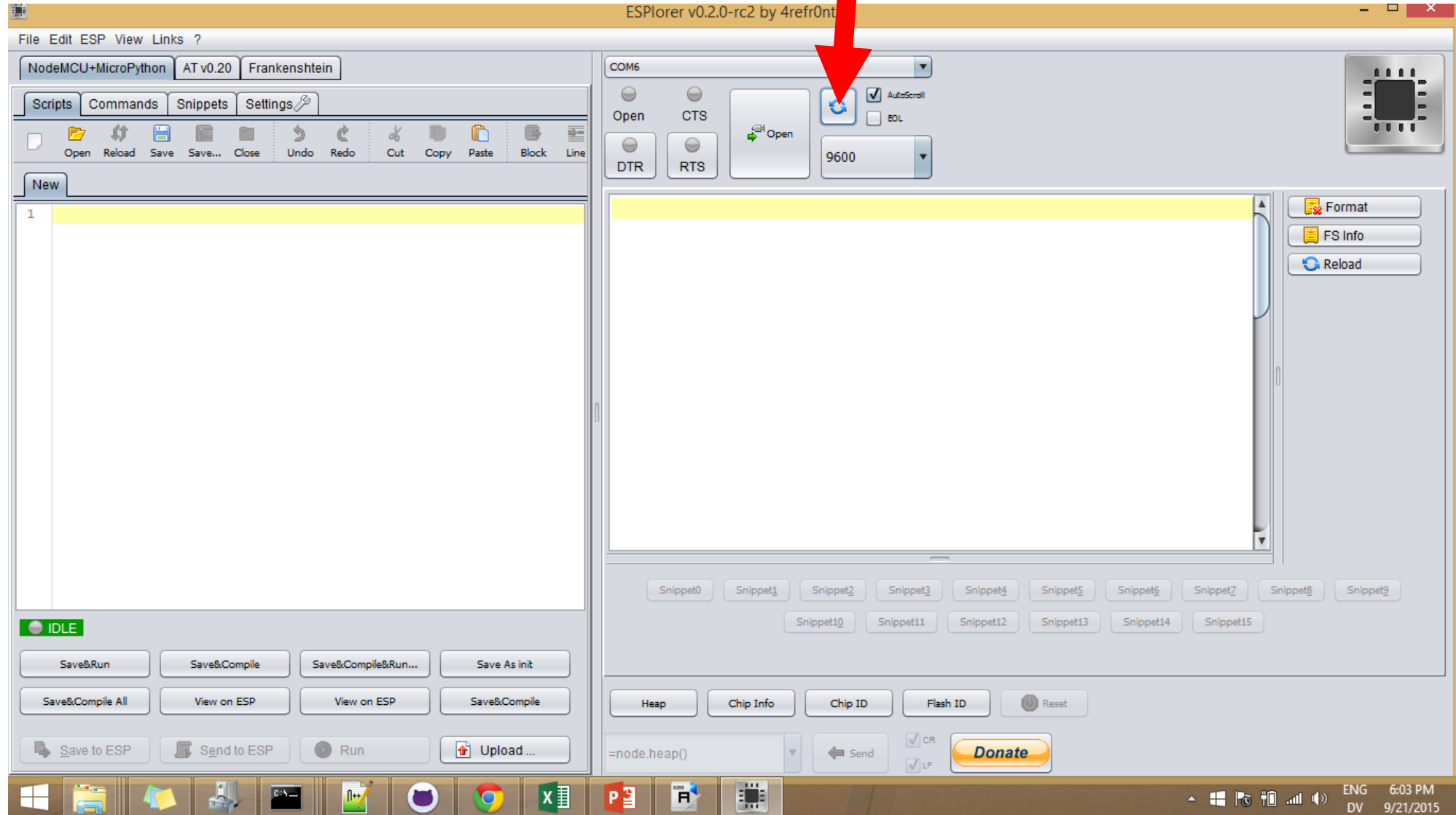
You should now reset the module.



# Open ESPlorer.jar



Click the refresh button, and it should detect your Esp8266 if it was flashed correctly and is plugged in. You may need to choose the COM port from the dropdown.



Click 'open', the response should look something like this.

The screenshot shows the ESP8266 IDE interface. The top menu bar includes File, Edit, ESP, View, and Links. Below the menu, there are tabs for 'NodeMCU+MicroPython', 'AT v0.20', and 'Frankenshtein'. A toolbar contains icons for Open, Reload, Save, Save..., Close, Undo, Redo, Cut, Copy, Paste, Block, and Line. The main editor area is currently empty, with a 'New' button above it. On the right side, there is a control panel for the serial port, showing 'COM6' selected. It includes buttons for 'Open', 'CTS', 'DTR', 'RTS', and 'Close' (highlighted by a red arrow). There are also checkboxes for 'AutoScroll' and 'EOL', and a baud rate dropdown set to '9600'. Below the control panel is a terminal window displaying the following text: 'PORT OPEN 9600', 'Communication with MCU...', 'Got answer! AutoDetect firmware...', and 'Can't autodetect firmware, because proper answer not received.' The last line is highlighted in pink. To the right of the terminal are buttons for 'Format', 'FS Info', and 'Reload'. At the bottom of the IDE, there are buttons for 'Snippet0' through 'Snippet15', 'Heap', 'Chip Info', 'Chip ID', 'Flash ID', and 'Reset'. A text input field contains '=node.heap()' and buttons for 'Send', 'CR', and 'LF'. A 'Donate' button is also present.

Esplorer v0.2.0-rc2 by 4refr0nt

File Edit ESP View Links ?

NodeMCU+MicroPython AT v0.20 Frankenshtein

Scripts Commands Snippets Settings

Open Reload Save Save... Close Undo Redo Cut Copy Paste Block Line

New

1

PORT OPEN 9600

Communication with MC...  
Got answer! AutoDetected firmware...

Can't autodetect firmware, because proper answer not received.

```
> =print('hello world')  
hello world  
>
```

Format FS Info Reload

Snippet0 Snippet1 Snippet2 Snippet3 Snippet4 Snippet5 Snippet6 Snippet7 Snippet8 Snippet9  
Snippet10 Snippet11 Snippet12 Snippet13 Snippet14 Snippet15

Heap Chip Chip ID Flash ID Reset

=print("hello world") Send CR LF Donate

IDLE

Save&Run Save&Compile Save&Compile&Run... Save As init

Save&Compile All View on ESP View on ESP Save&Compile

Save to ESP Send to ESP Run Upload ...

ENG 6:05 PM  
DV 9/21/2015

You can enter commands down here, then click 'send'.  
To do the canonical 'hello world', type  
=print('hello world')  
and click 'send'



Click 'reload' to list the files on the device

ESPlorer v0.2.0-rc2 by 4refr0nt

File Edit ESP View Links ?

NodeMCU+MicroPython AT v0.20 Frankenshtein

Scripts Commands Snippets Settings

Open Reload Save Save... Close Undo Redo Cut Copy Paste Block Line

New

```
> =print('hello world')
hello world
>
-----
sendToTS.lc      : 3224 bytes
sendToTS.lua    : 4751 bytes
-----
Total file(s)   : 2
Total size      : 7975 bytes

> r,u,t=file.fsinfo() print("Total : "..t.." bytes\r\nUsed : "..u.." bytes\r\nRemain: ..r.. bytes")
Total : 3396281 bytes
Used : 15311 bytes
Remain: 3380970 bytes
>
```

Format Info Reload sendToTS.lc sendToTS.lua

Snippet0 Snippet1 Snippet2 Snippet3 Snippet4 Snippet5 Snippet6 Snippet7 Snippet8 Snippet9 Snippet10 Snippet11 Snippet12 Snippet13 Snippet14 Snippet15

Heap Chip Info Chip ID Flash ID Reset

=print('hello world') Send CR LF Donate

ENG DV 6:07 PM 9/21/2015

Click 'heap' or send '=node.heap()' to get the heap size (free memory).

ESPlorer v0.2.0-rc2 by 4refr0nt

File Edit ESP View Links ?

NodeMCU+MicroPython AT v0.20 Frankenshtein

Scripts Commands Snippets Settings

Open Reload Save Save... Close Undo Redo Cut Copy Paste Block Line

New

1

COM6

Open CTS Close AutoScroll EOL

DTR RTS 9600

```
> =print('hello world')
hello world
>
-----
sendToTS.lc : 3224 bytes
sendToTS.lua : 4751 bytes
-----
Total file(s) : 2
Total size : 7975 bytes

> r,u,t=file.fsinfo() print("Total : \"..t..\" bytes\r\nUsed : \"..u..\" bytes\r\nRemain: \"..t..\" bytes")
Total : 3396281 bytes
Used : 15311 bytes
Remain: 3380970 bytes

>
```

Snippet0 Snippet1 Snippet2 Snippet3 Snippet4 Snippet5 Snippet6 Snippet7 Snippet8 Snippet9 Snippet10 Snippet11 Snippet12 Snippet13 Snippet14 Snippet15

Heap Chip Info Chip ID Flash ID Reset

=print('hello world') Send CR LF Donote

IDLE

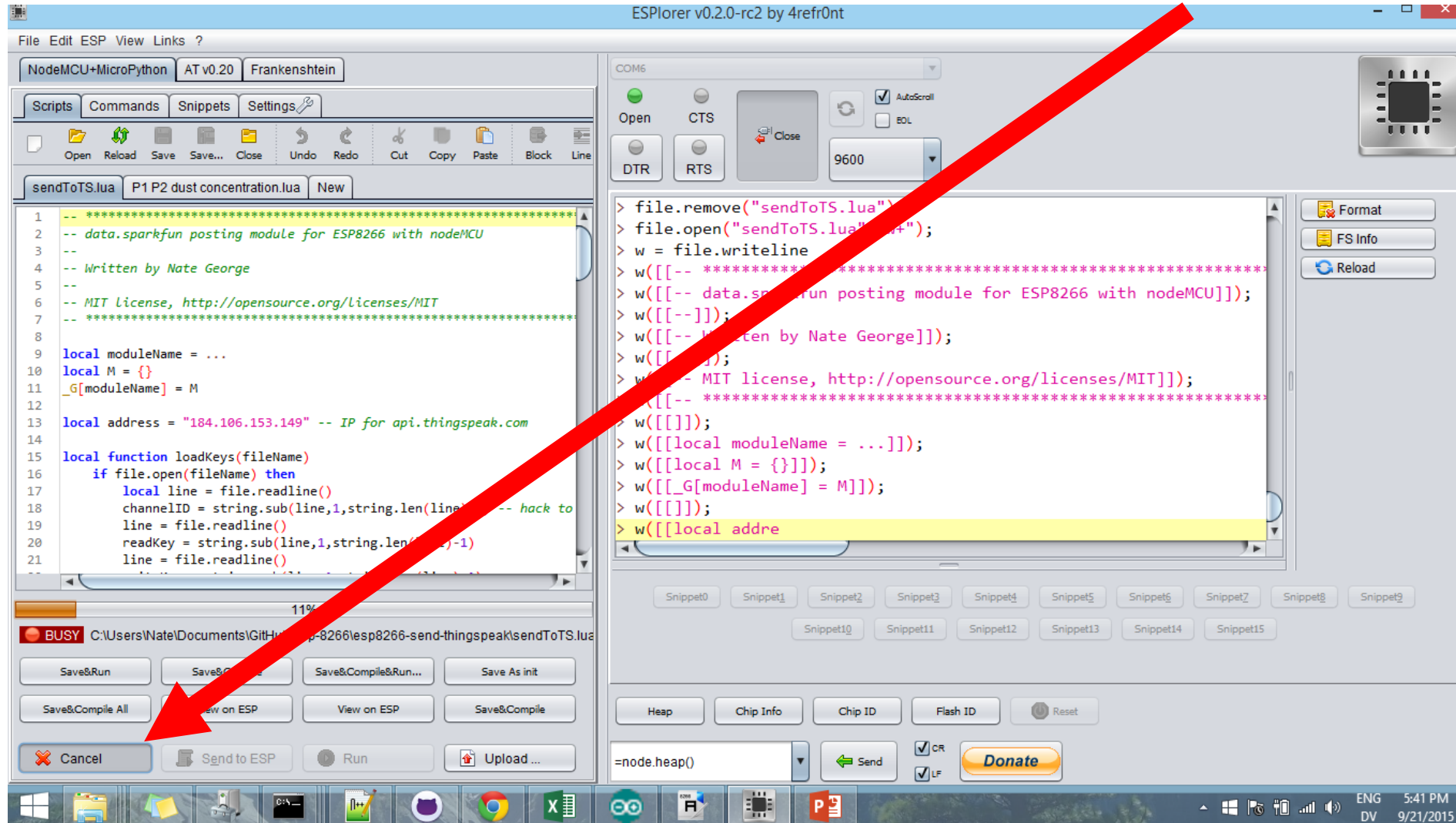
Save&Run Save&Compile Save&Compile&Run... Save As init

Save&Compile All View on ESP View on ESP Save&Compile

Save to ESP Send to ESP Run Upload ...

ENG 6:07 PM 9/21/2015

Click 'save to ESP' to write your file to the device. Clicking 'SaveAndCompile' will compile it, and save a lot of memory at runtime.



The programming is done in the Lua language, which is C-like. I made a few chunks of code for using NodeMCU:

Wifi network chooser:

<https://github.com/wordsforthewise/ESP-8266-network-connect>

Makes a server at 192.168.4.1 and asks for wifi credentials, then logs into the network.

Send data to sparkfun:

<https://github.com/wordsforthewise/esp8266-send-to-sparkfun>

Send data to thingspeak:

<https://github.com/wordsforthewise/esp8266-send-thingspeak>

