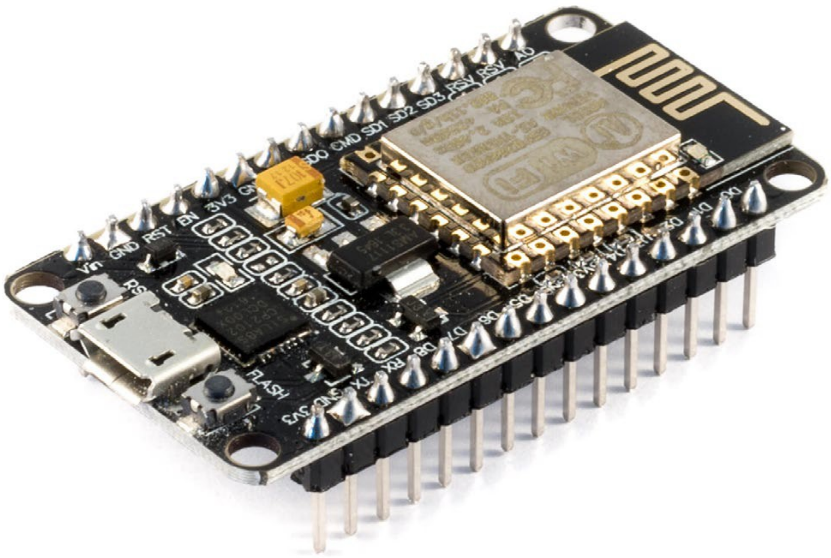
Welcome!

And thank you for purchasing our **AZ-Delivery NodeMCU V2 Amica**! On the following pages, we will take you through the first steps of the installation process to the first script.

We wish you a lot of fun!



https://az-delivery.de/nodemcu-v2-amica

The **AZ-Delivery NodeMCU V2 Amica** has many improvements compared to its previous version. The new USB interface provides greater driver compatibility and a more comfortable transfer of firmware and codes, without having to press the reset and flash buttons at the right moment. In addition, the narrower board design is suitable for a Breadboard.

Overview of the most important information

» Programming via micro USB-B-cable

» Power supply via:
  » Micro USB-B on the USB-port of the computer
  » Micro USB-B on the 5V USB-power adapter

» 11 digital I / O-Pins (3,3V!)
» 1 analog I / O-Pin
» ESP-12E processor with ESP8266 WLAN-module
» CP2102 USB-interface
» Programmable via Arduino Code and Lua

On the following pages, you will find information about
» **Driver installation and preparation of the Arduino IDE**,
Instructions for
» **the first script by Arduino Code**,
followed by
» **System preparation for working with Lua**
And a guide for
» **the first Lua-Script**.

Overview of all links

Driver:

» *http://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers*

Lua-Services:

» Firmware-Generator: *https://nodemcu-build.com/*

» esptool.py: *https://github.com/espressif/esptool*

» NodeMCU-Flasher: *https://github.com/nodemcu/nodemcu-flasher/blob/master/Win32/Release/ESP8266Flasher.exe*

» Esplorer: *http://esp8266.ru/esplorer/*

» Luatool: *https://github.com/4refr0nt/luatool*

» Lua-Tutorialscript – Listing WLAN Access Points: *https://raw.githubusercontent.com/pradeesi/NodeMCU-WiFi/master/list_ap.lua*

Other Tools:

» Python: *https://www.python.org/downloads/*

Interesting information from AZ-Delivery

» AZ-Delivery G+Community: *https://plus.google.com/communities/115110265322509467732*

» AZ-Delivery on Facebook: *https://www.facebook.com/AZDeliveryShop/*

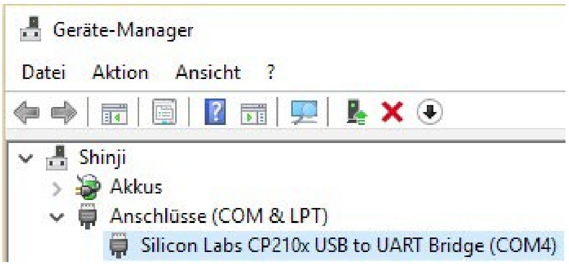
Driver installation

The **AZ-Delivery NodeMCU V2 Amica** connects to your computer via a Micro-USB-cable. The microcontroller uses a **CP2102-Chip** for the USB interface, which is normally automatically recognized by Windows and partly recognized by MacOS systems.

If that is not the case, then please download the latest driver from this link, and then unzip it:

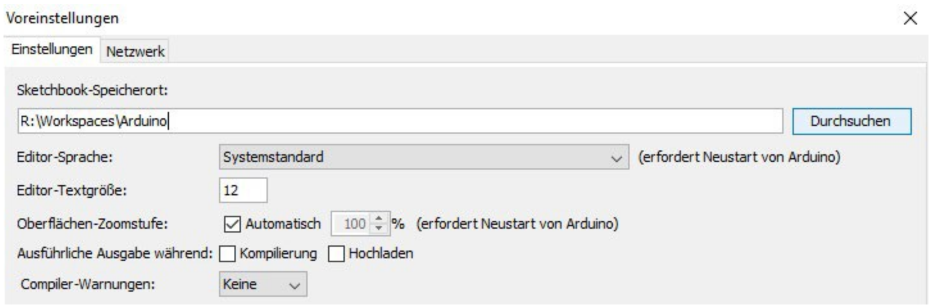
» *http://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers*

On Windows, you can simply install it by running "**CP210xVCPInstaller_x86.exe**" or "**CP210xVCPInstaller_x64.exe**", depending on your system. As a Mac user, you should install the DMG file, which is located in your loaded archive.
After you reconnect the NodeMCU, it should be recognized as a "**Silicon Labs CP210x USB to UART Bridge**" device (Windows).



Preparation of the Arduino IDE

Visit the following webpage *https://www.arduino.cc/en/Main/Software* and download the latest version for your operating system. Alternatively, you can register for the Arduino Web-Editor and follow the easy-to-understand installation instructions that are provided there. The following first



Voreinstellungen

Einstellungen | Netzwerk

Sketchbook-Speicherort: R:\Workspaces\Arduino

Editor-Sprache: Systemstandard (erfordert Neustart von Arduino)

Editor-Textgröße: 12

Oberflächen-Zoomstufe: [✓ Automatisch] 100 % (erfordert Neustart von Arduino)

Ausführliche Ausgabe während: ☐ Kompilierung ☐ Hochladen

Compiler-Warnungen: Keine

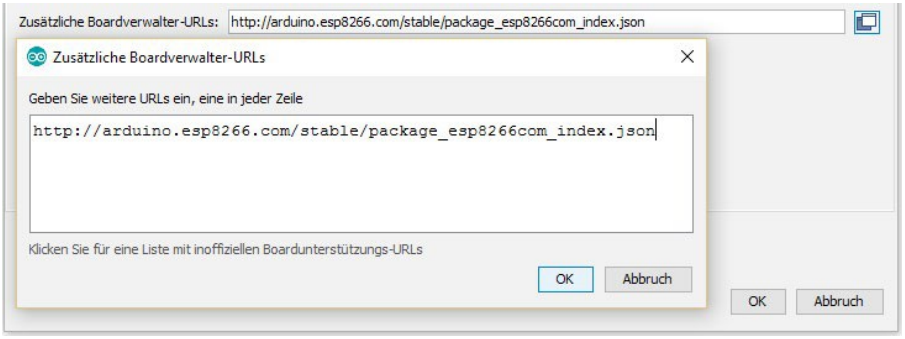
steps will use the desktop variations for Windows.

If the program has already been started, then the storage location for your first sketchbook should be under **file > preferences**, for example under **my documents \ Arduino**. That way, your Arduino scripts, named as "**Sketche**", will be stored at a place of your choice.

The NodeMCU, however, is not part of the standard repertoire of the IDE, for this reason, the Board manager needs to be expanded. In the same window, under “Additional Board Administrator URLs” add the following address:

»

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



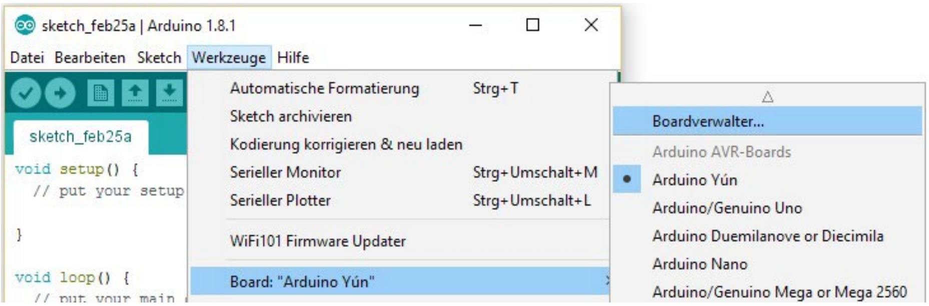
Zusätzliche Boardverwalter-URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

Zusätzliche Boardverwalter-URLs

Geben Sie weitere URLs ein, eine in jeder Zeile

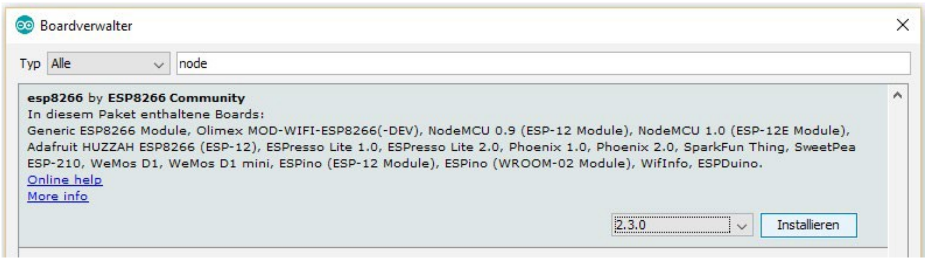
```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

Klicken Sie für eine Liste mit inoffiziellen Boardunterstützungs-URLs


sketch_feb25a | Arduino 1.8.1
Datei Bearbeiten Sketch Werkzeuge Hilfe
Automatische Formatierung Strg+T
Sketch archivieren
Kodierung korrigieren & neu laden
Serieller Monitor Strg+Umschalt+M
Serieller Plotter Strg+Umschalt+L
WiFi101 Firmware Updater
Board: "Arduino Yún"
Boardverwalter...
Arduino AVR-Boards
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560

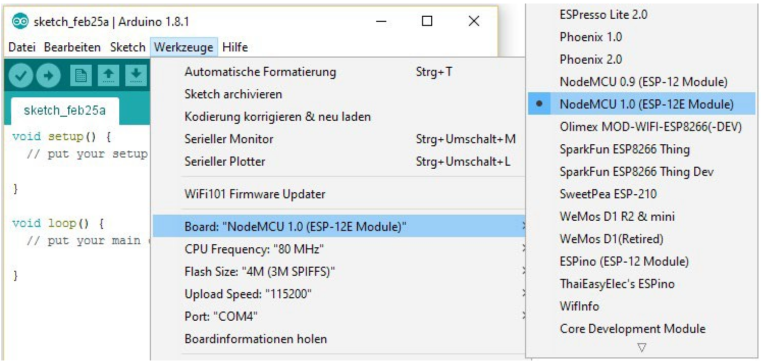
```
void setup() {
  // put your setup

}

void loop() {
  // put your main
```

Once completed, got to **Tools** > **Board** > **Board Administrator** and install the Board library "**esp8266 by ESP8266 Community**".



esp8266 by **ESP8266 Community**
In diesem Paket enthaltene Boards:
Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifInfo, ESPDuino.

Now you can choose the correct board, named "**NodeMCU 1.0 (ESP12E Module)**", in addition, a CPU frequency of 80 MHz, the memory size "**4M (3M SPIFFS)**", a baud rate of e.g. **115200** and the appropriate port ("**COM**" for Windows, "**ttyUSB**" for MacOS).


sketch_feb25a | Arduino 1.8.1
Datei Bearbeiten Sketch Werkzeuge Hilfe
Automatische Formatierung Strg+T
Sketch archivieren
Kodierung korrigieren & neu laden
Serieller Monitor Strg+Umschalt+M
Serieller Plotter Strg+Umschalt+L
WiFi101 Firmware Updater
Boardinformationen holen
ESPresso Lite 2.0
Phoenix 1.0
Phoenix 2.0
NodeMCU 0.9 (ESP-12 Module)
NodeMCU 1.0 (ESP-12E Module)
Olimex MOD-WIFI-ESP8266(-DEV)
SparkFun ESP8266 Thing
SparkFun ESP8266 Thing Dev
SweetPea ESP-210
WeMos D1 R2 & mini
WeMos D1(Retired)
ESPino (ESP-12 Module)
ThaiEasyElec's ESPino
WifInfo
Core Development Module

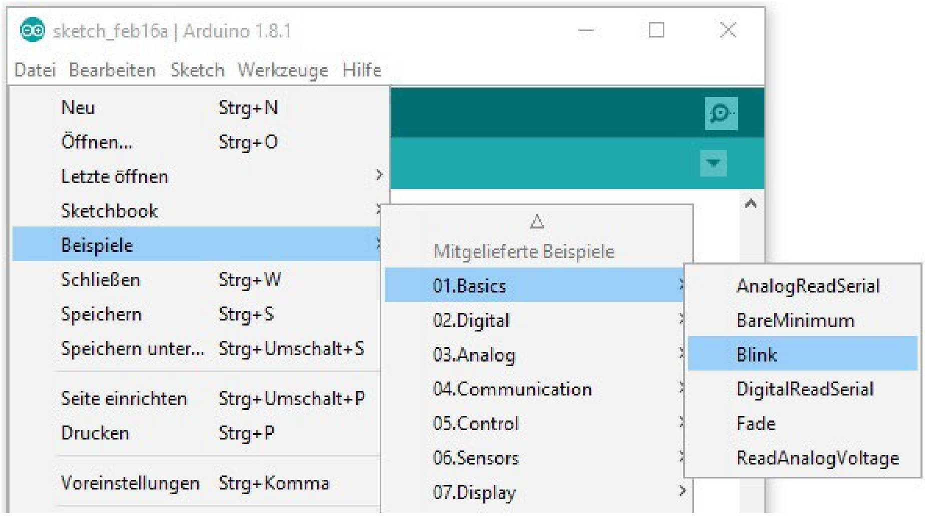
Board: "NodeMCU 1.0 (ESP-12E Module)"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "115200"
Port: "COM4"

```
void setup() {
  // put your setup

}

void loop() {
  // put your main

}
```

The first script by Arduino Code

Although the first sign of success in most programming languages is the phrase: "Hello World!", for Arduino, the first sign of success is the blinking of the board's internal LED. In correspondence, the script is called "**Blink**".

» Start the Arduino IDE and open under "**Start**" the Blink-Script.

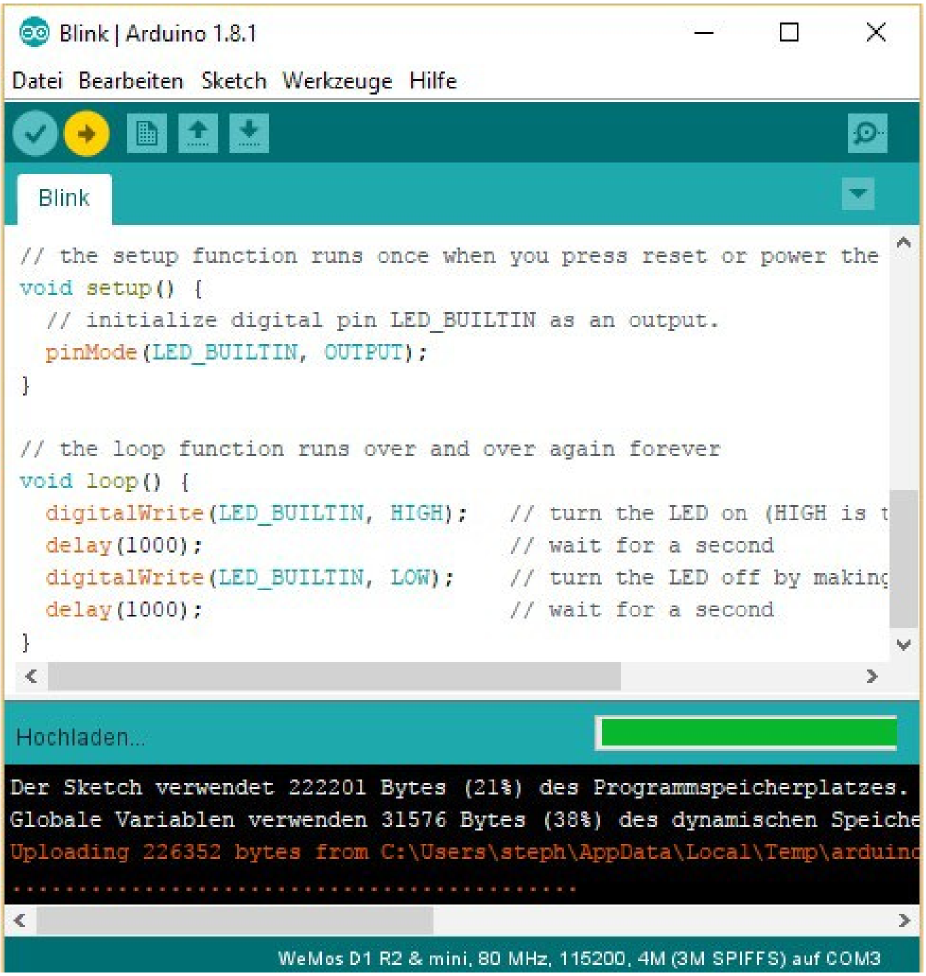


Each sketch always contains the "**setup**" and "**loop**" method. The "setup" method is initially executed and is typically used to initialize pins and separated hardware. The "loop" method is then permanently repeated, and thus contains almost all other functions.

The board's internal LED has been automatically selected for some time via the IDE's own variable "**LED_BUILTIN**". While it compiles an I/O Pin on Arduino, it is addressed to pin "**16**" on the NodeMCU, even though there are only eleven digital I/O Pins. It can also be addressed via "**D0**":

LED_BUILTIN == 16 == D0

The second icon below the command bar will load the sketch into the NodeMCU.


Blink | Arduino 1.8.1
Datei Bearbeiten Sketch Werkzeuge Hilfe
Blink
Hochladen...
WeMos D1 R2 & mini, 80 MHz, 115200, 4M (3M SPIFFS) auf COM3

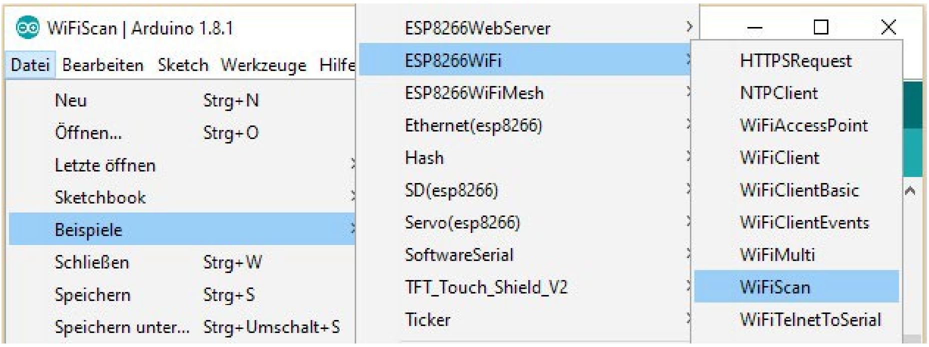
```
// the setup function runs once when you press reset or power the
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is t
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by makin
  delay(1000);                       // wait for a second
}
```

```
Der Sketch verwendet 222201 Bytes (21%) des Programmspeicherplatzes.
Globale Variablen verwenden 31576 Bytes (38%) des dynamischen Speiche
Uploading 226352 bytes from C:\Users\steph\AppData\Local\Temp\arduino
.......................................
```

If the upload was successful, then the LED of your **NodeMCU** will begin to blink every second.

You did it! Congratulations!

Next, you should try the special feature of the NodeMCU, namely the WLAN module.
Load the "**WiFiScan**" sketch on your board and then start the Serial Monitor with the correct baud rate. After that, in a few seconds, you should see a list with all of the available WLAN-Access-Points, in your surroundings, as well as their respective signal strength.



With the help of the Arduino Code, you can do and achieve so much more with a NodeMCU. Start your search for more possibilities by looking at other example sketches, which you will find in the Arduino library and on the internet. For example, here at *http://michael- sarduino.blogspot.de/search?q=8266*. For hardware support, our online store is always at your disposal:

https://az-delivery.de

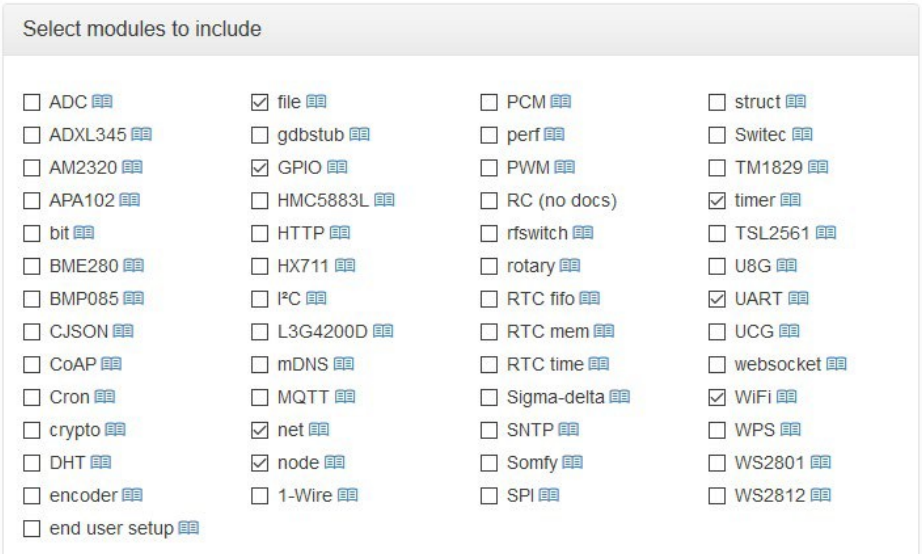
If you would like to go ahead and learn how to use the NodeMCU with Lua scripts, then please continue reading.

System preparation for working with LUA

The **NodeMCU V2 Amica** normally comes with an AT firmware, from the manufacturer's AI-Thinker. In order to use the chipset with LUA's script language, then you would have to first create the foundation for that. To successfully complete that, you would have to put together the appropriate firmware for your project:

» *https://nodemcu-build.com/*

In addition to choosing the stable or developer's version, there are also plenty of options to choose from that would allow the extension of the functionality of your board. Too many unnecessary extensions, however, would only slow down the NodeMCU. For our Tutorial script, the default specifications are



Select modules to include

☐ ADC	☑ file	☐ PCM	☐ struct
☐ ADXL345	☐ gdbstub	☐ perf	☐ Switec
☐ AM2320	☑ GPIO	☐ PWM	☐ TM1829
☐ APA102	☐ HMC5883L	☐ RC (no docs)	☑ timer
☐ bit	☐ HTTP	☐ rfswitch	☐ TSL2561
☐ BME280	☐ HX711	☐ rotary	☐ U8G
☐ BMP085	☐ I²C	☐ RTC fifo	☑ UART
☐ CJSON	☐ L3G4200D	☐ RTC mem	☐ UCG
☐ CoAP	☐ mDNS	☐ RTC time	☐ websocket
☐ Cron	☐ MQTT	☐ Sigma-delta	☑ WiFi
☐ crypto	☑ net	☐ SNTP	☐ WPS
☐ DHT	☑ node	☐ Somfy	☐ WS2801
☐ encoder	☐ 1-Wire	☐ SPI	☐ WS2812
☐ end user setup			

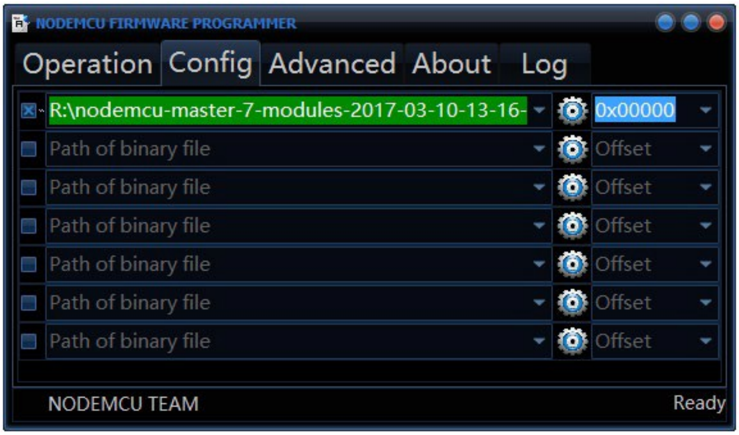
sufficient.

Just enter your e-mail address twice in the first block and then click on "**Start your build**". In the following minutes, you will receive an order confirmation and an e-mail with links, from which you can download the firmware. There is an integer and a Float-Version to choose from. The only difference is that the latter can handle floating-point numbers. Which variation you choose, is irrelevant for our tutorial.

To install the firmware, a Flash tool is required, similar to the system independent Python script "**esptool.py**". On Windows, more comfortable will be the use of the "**NodeMCU-Flasher**" program, which you can download from here:

» *https://github.com/nodemcu/nodemcu-flasher/blob/master/Win32/Release/ESP8266Flasher.exe*

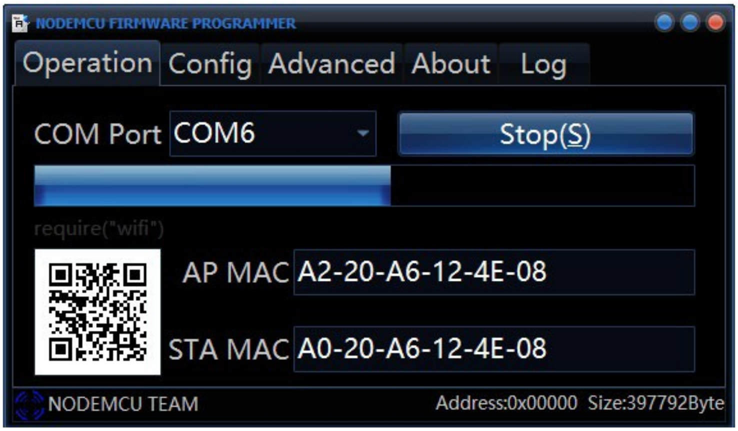
Start the program and select under "**Config**" your already downloaded firmware. Leave the address at **0x00000**.



R:\nodemcu-master-7-modules-2017-03-10-13-16- 0x00000
Path of binary file Offset
Path of binary file Offset
Path of binary file Offset
Path of binary file Offset
Path of binary file Offset
Path of binary file Offset

NODEMCU TEAM Ready

Under "**Advanced**" you will find fine adjustments for the board. For us, the default settings: baud rate of **115200**, **4 MB** flash memory, **40 MHz** memory speed and the "**DIO**"-SPI-mode are satisfactory. Then start the Flash-process for the COM-Port of your connected NodeMCU and wait for the green tick to appear at the bottom left.



COM Port COM6

require("wifi")

AP MAC A2-20-A6-12-4E-08

STA MAC A0-20-A6-12-4E-08

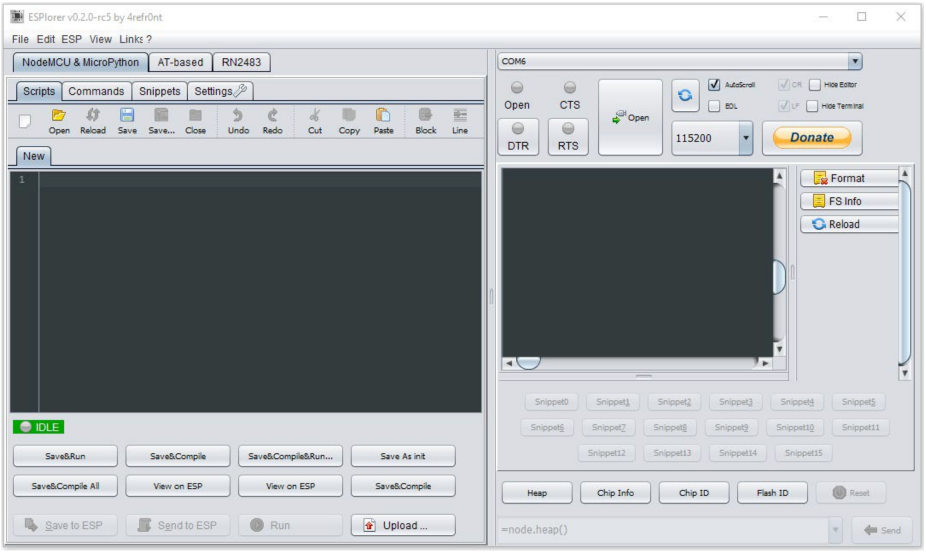
Address:0x00000 Size:397792Byte

Finally, you would need a tool, with which you can write your Lua scripts and, above all, load them to the NodeMCU. Pure console usage offers the "**Luatool**" package.

The "**Esplorer**" is also platform independent and with its graphical user interface, is one of the most liked and preferred variants. We would also use it for the tutorial:

» Esplorer: *http://esp8266.ru/esplorer/*

Download the appropriate version for your operating system and start the "**Esplorer.bat**" (Windows) after you unzip and unpack the archive.



As you can see, the program comes with some predefined commands and has the ability to operate alongside other systems, besides the NodeMCU. If we had not flashed a new firmware in advance, now we would have been limited to use only the commands under "**AT-based**" tab.

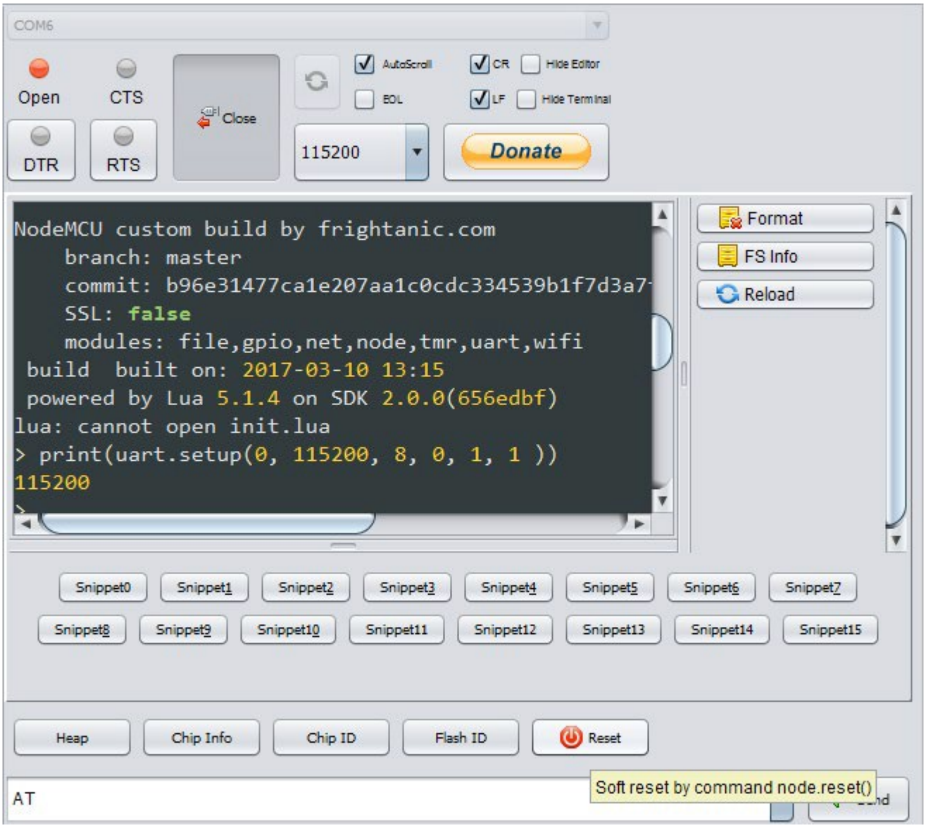
The first LUA script

Such as "Hello World" script, a similar function should be used for the NodeMCU, as with "**WiFiScan**" for the Arduino IDE. The complete code for this can be copied from the following link:

» *https://raw.githubusercontent.com/pradeesi/NodeMCU-WiFi/master/list_ap.lua*

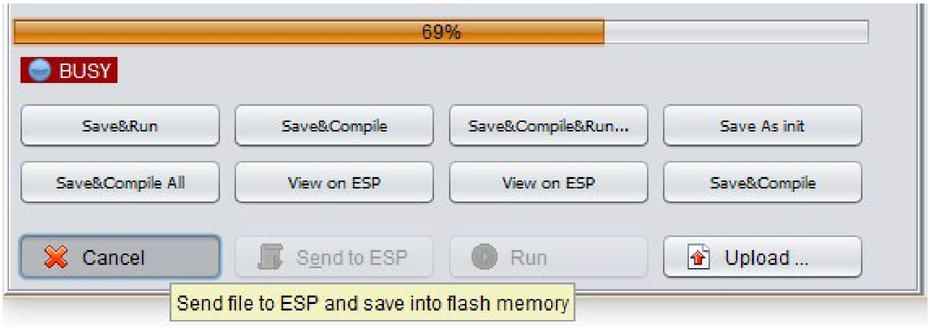
Firstly, you should check if the installation of the firmware from the previous step has been successful. In order to that, you have to choose the correct COM port (from the right side) and the correct baud rate (for us **115200**), then click on "**Open**" (on the right side).

On the terminal “**Communication with MCU..**" will be displayed. Now press the reset button on the NodeMCU, and the board will run the boot routine, indicating the information about the installed firmware. That should look similar to this screenshot:



```
NodeMCU custom build by frightanic.com
    branch: master
    commit: b96e31477ca1e207aa1c0cdc334539b1f7d3a7
    SSL: false
    modules: file,gpio,net,node,tmr,uart,wifi
 build  built on: 2017-03-10 13:15
 powered by Lua 5.1.4 on SDK 2.0.0(656edbf)
lua: cannot open init.lua
> print(uart.setup(0, 115200, 8, 0, 1, 1 ))
115200
>
```

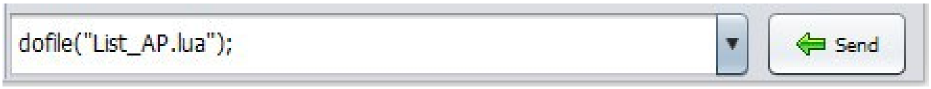
You should now copy the example code, located in the dark window on the left side of the program, under the "**Scripts**" tab. If you would like to make a shortcut or are simply impatient, then click on "**Send to ESP**". The code is then executed line by line on the NodeMCU, and the result is distributed on the terminal, without having to store the script on the board.



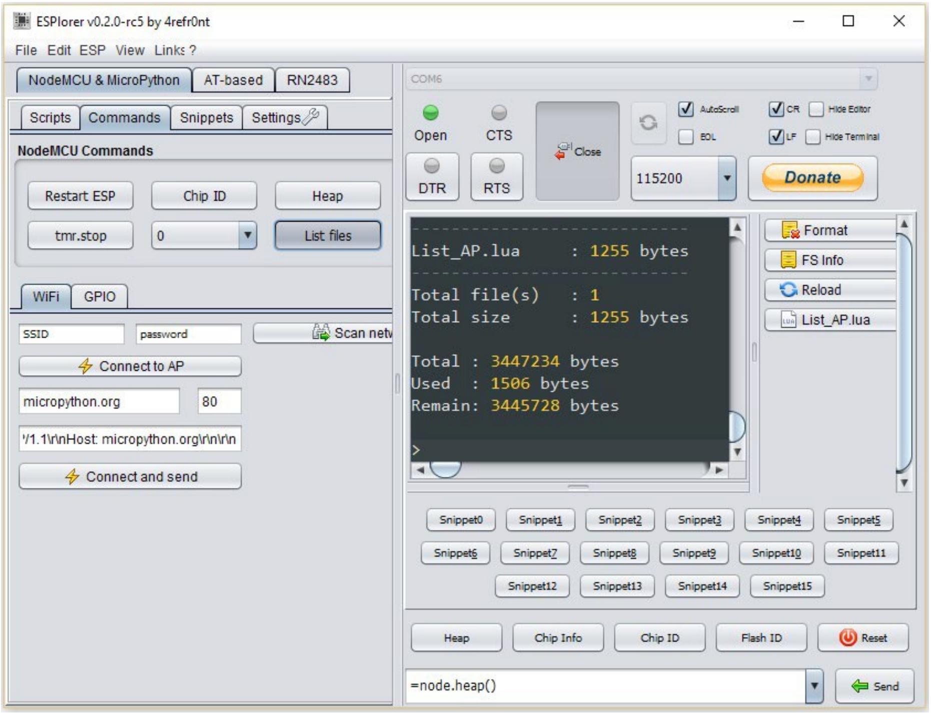
To save, click on "**Save**" and give the script a name, e.g. "**List_AP.lua**". It is then automatically loaded and listed on the NodeMCU. If that does not happen, then click on "**Save to ESP**", which is located in the lower left corner.

To check if the file is now on the board, go to the "**Commands**" area and click on "**List files**". The "**List_AP.lua**" file should now be listed in the terminal.
The script can also be started from the command line, located at the bottom right with the following command, directly from the **NodeMCU**.

```
» dofile("List_AP.lua");
```



```
dofile("List_AP.lua");
```



```
-----------------------------
List_AP.lua      : 1255 bytes
-----------------------------
Total file(s)    : 1
Total size       : 1255 bytes

Total : 3447234 bytes
Used  : 1506 bytes
Remain: 3445728 bytes

>
```

Now it is time to learn. You can do that with the help of many example scripts and other tutorials, which you can find on the internet. Here *http://nodemcu.com/index_en.html#fr_5475f7667976d8501100000f* you can begin your search.

And for more hardware, our online store is always at your disposal:

https://az-delivery.de

Enjoy!

Imprint

https://az-delivery.de/pages/about-us