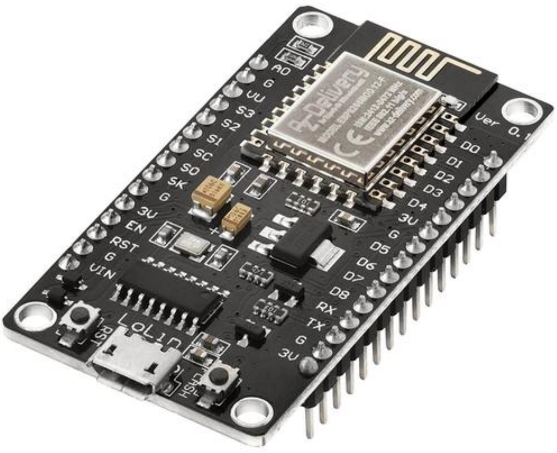
AZ-Delivery
Ihr Experte für Mikroelektronik!

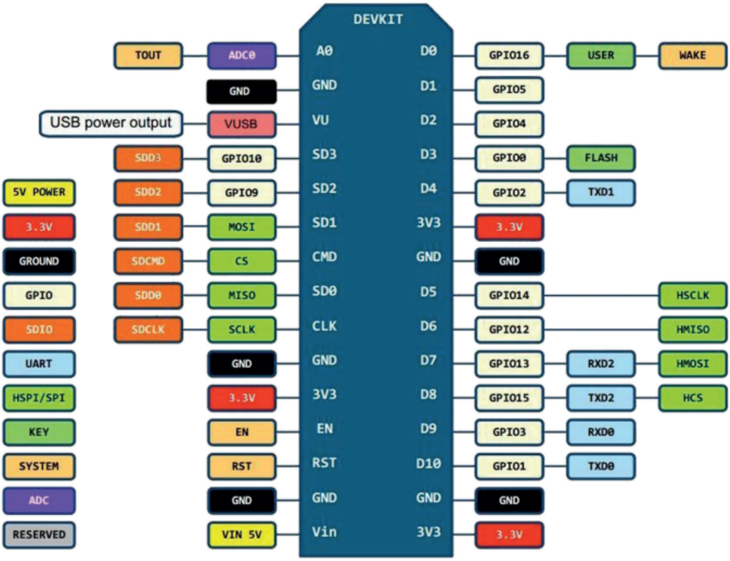
Welcome!

And thank you for purchasing our **AZ-Delivery NodeMCU Lolin V3**! 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!



The **AZ-Delivery NodeMCU Lolin V3** has many improvements compared to its predecessor. Moreover, the new USB interface provides better driver compatibility and a more comfortable transfer of firmware and code, without having to press the reset and flash buttons at the right moment. The NodeMCU is powered by a micro-USB port.



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
- » CH340 USB interface
- » Programmable via Code and Lua

On the following pages, you will find information about

» ***driver installation and preparation of the Arduino IDE****,*

A guide for

» ***the first script by Code***,

followed by

» ***system preparation for working with Lua***

And instructions for

» ***the first Lua script***.

Overview of all links

Driver:

» Windows: *http://www.wch.cn/download/CH341SER_ZIP.html*

» Mac: *http://www.wch.cn/download/CH341SER_MAC_ZIP.html*

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/

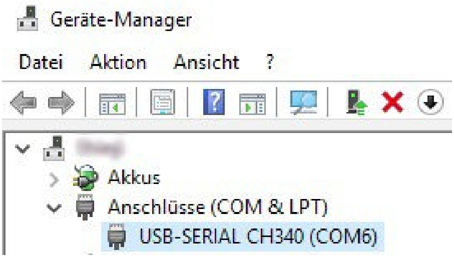
Driver installation

The **AZ-Delivery NodeMCU Lolin V3** connects to your computer via a micro-USB cable. As with most AZ-Delivery boards, a **CH340-Chip** is used for communication, which is automatically detected and recognized by Windows.

If that is not the case, then please download the latest driver from the links below, and then unzip it.

» Windows:

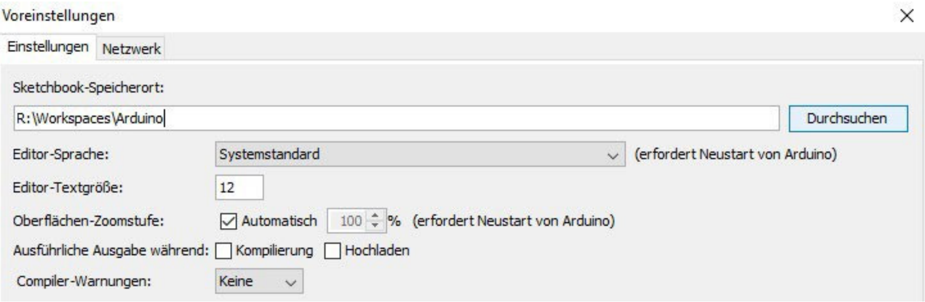
On Windows, you can simply install it by running "**SETUP.EXE**" located in the "**CH341SER**" folder. Mac users are best advised to follow the installation instructions that come with the driver's package.
After reconnecting the Nanos, it should be recognized as a "**USB-SERIAL CH340**" device (Windows).



[redacted]
Akkus
Anschlüsse (COM & LPT)
USB-SERIAL CH340 (COM6)

Preparation of the Arduino IDE

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



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

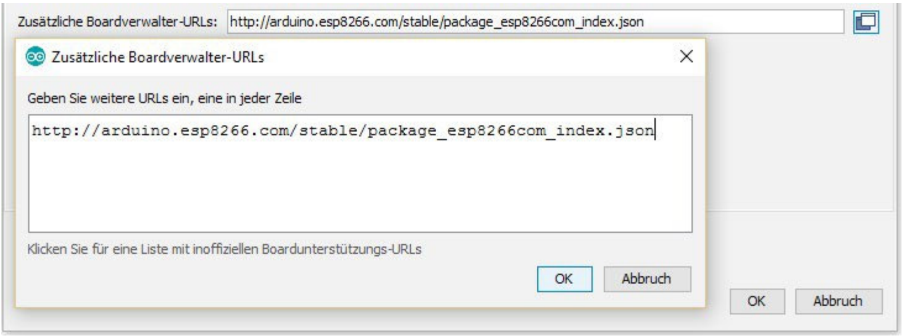
Compiler-Warnungen: Keine

If the program has already been started, then the storage location of your first sketchbook should be under ***file** > **preferences,*** for example, under ***my documents***. That way your scripts, namely "***Sketche***", will be saved in the place you want.

However, the NodeMCU is not part of the standard repertoire of the IDE, for this reason, the board manager needs to be expanded. In the same window, add the following link under "Additional Board Administrator URLs":

»

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



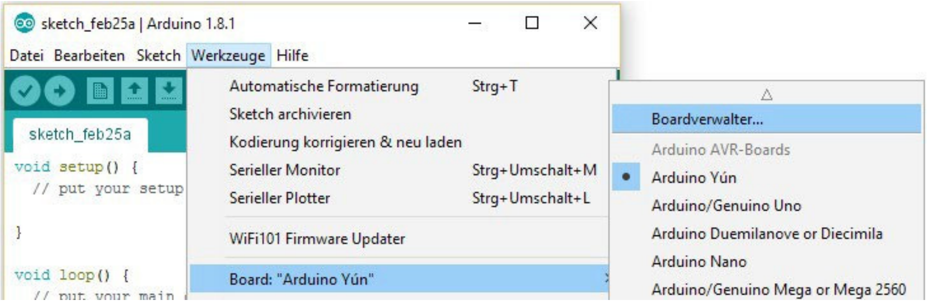
Zusätzliche Boardverwalter-URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

Zusätzliche Boardverwalter-URLs

Geben Sie weitere URLs ein, eine in jeder Zeile

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

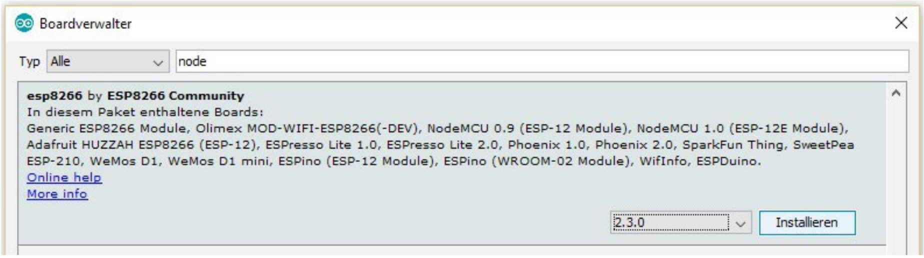
Klicken Sie für eine Liste mit inoffiziellen Boardunterstützungs-URLs


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
sketch_feb25a

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

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

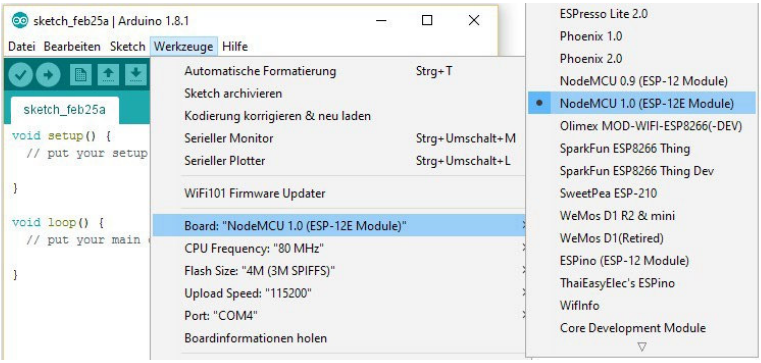
Once you have completed that, go to **Tools** > **Board** > **Board Administrator** and install the board library "**esp8266 by ESP8266 Community**".



node

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, namely "**NodeMCU 1.0 (ESP12E Module)**", in addition, select a CPU frequency of **80 MHz**, a memory size of "**4M (3M SPIFFS)**", a baud rate of e.g. **115200** and the appropriate port ("**COM**" on Windows, "**ttyUSB**" on 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
sketch_feb25a

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

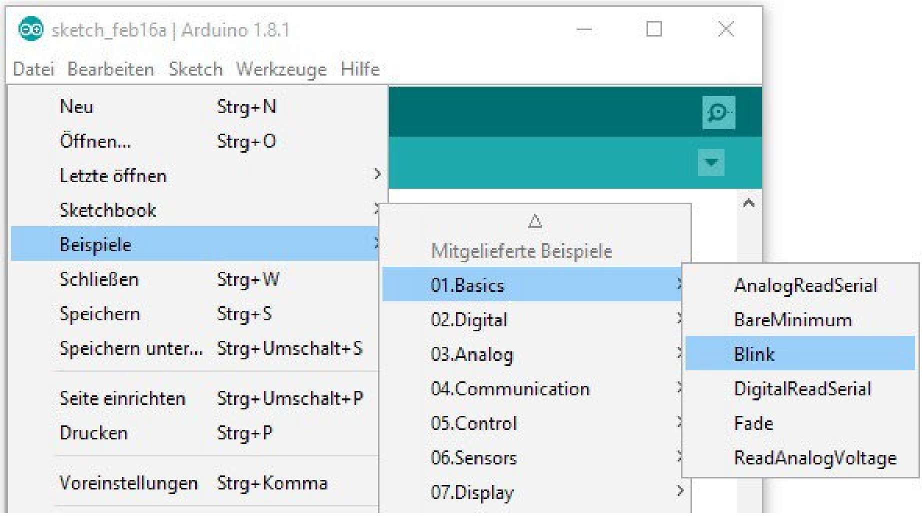
```
void setup() {
  // put your setup
}

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

The first script by Code

Even though the first sign of success in most programming languages is the "Hello World!" phrase, in the case of Arduinos, the first sign of success is the blinking of the board's internal LED. Accordingly, the name of the script is "***Blink***".

» Start the Arduino IDE and open the Blink script, located under "**Start**".

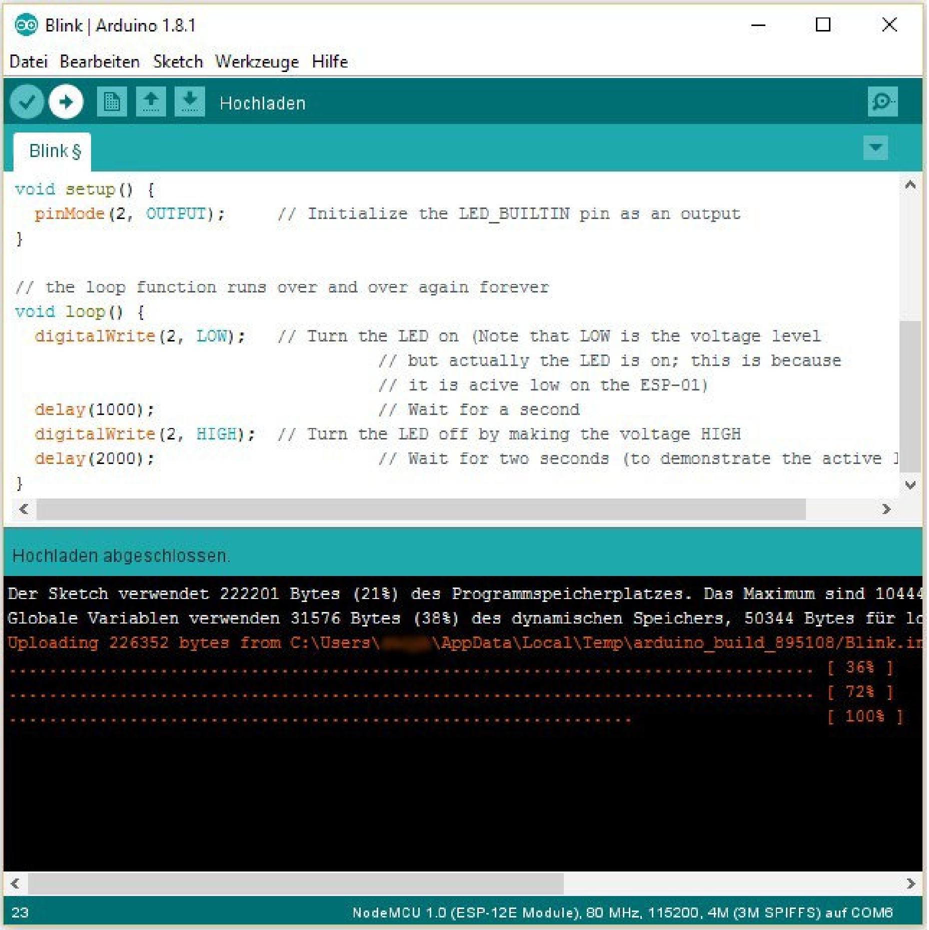


Each sketch always contains the "***setup***" and "***loop***" method. The "setup" method is initially executed and is typically used to initialize pins and separated hardware. The "loop" method is then permanently repeated, and thus contains almost all other functions.

The board's internal LED has been automatically selected for some time via the IDE's own variable "**LED_BUILTIN**". However, the library is designed for the Amica model, which is why the internal LED **GPIO16**, i.e. **Pin D0** is fixed. With the **Lolin V3**, however, it is due to "**GPIO2/D4**".

In the sketch, you should change all "**LED_BUILTIN**" to "**2**" or "**D4**" (please see the picture below).

With the second icon, below the command bar, you can load the sketch onto the NodeMCU.


Blink | Arduino 1.8.1
Datei Bearbeiten Sketch Werkzeuge Hilfe
Hochladen
Blink §
Hochladen abgeschlossen.
23
NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (3M SPIFFS) auf COM6

```
void setup() {
  pinMode(2, OUTPUT);     // Initialize the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(2, LOW);   // Turn the LED on (Note that LOW is the voltage level
                                    // but actually the LED is on; this is because
                                    // it is acive low on the ESP-01)
  delay(1000);                      // Wait for a second
  digitalWrite(2, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);                      // Wait for two seconds (to demonstrate the active l
}
```

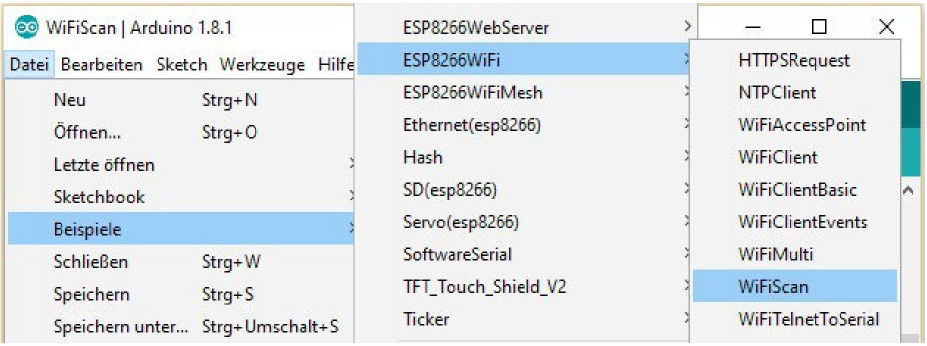
```
Der Sketch verwendet 222201 Bytes (21%) des Programmspeicherplatzes. Das Maximum sind 10444
Globale Variablen verwenden 31576 Bytes (38%) des dynamischen Speichers, 50344 Bytes für lo
Uploading 226352 bytes from C:\Users\[redacted]\AppData\Local\Temp\arduino_build_895108/Blink.in
................................................................................ [ 36% ]
................................................................................ [ 72% ]
.............................................................                    [ 100% ]
```

If the upload was successful, then the LED of your **NodeMCU** will start blinking every second.

You did it! Congratulations!

Next, you should try the special feature of the NodeMCU, namely the WLAN module.

Load the "**WiFiScan**" sketch onto your board, and then start the serial monitor with the correct baud rate. A few seconds later, you should see all of the WLAN-Access-Points that are available in your surroundings, as well as their respective signal strength.



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

https://az-delivery.de

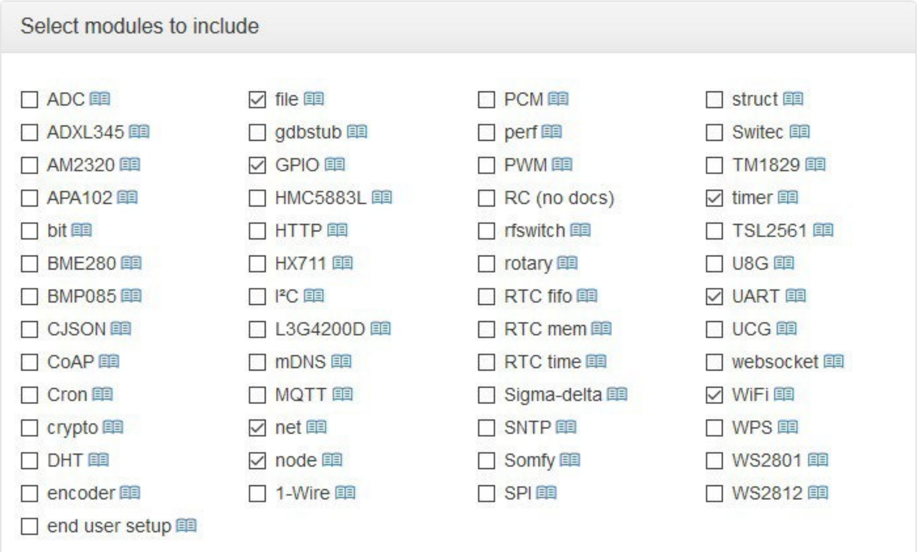
But if you want to go ahead and learn how to use the NodeMCU with Lua scripts, then continue reading.

System preparation for working with LUA

The **NodeMCU Lolin V3** normally comes with an AT firmware, from the manufacturer's AI-Thinker. If you want to be able to use the chipset with the LUA's script language, first you must create the foundation. For that you have to put the corresponding firmware together 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 sufficient.



Select modules to include

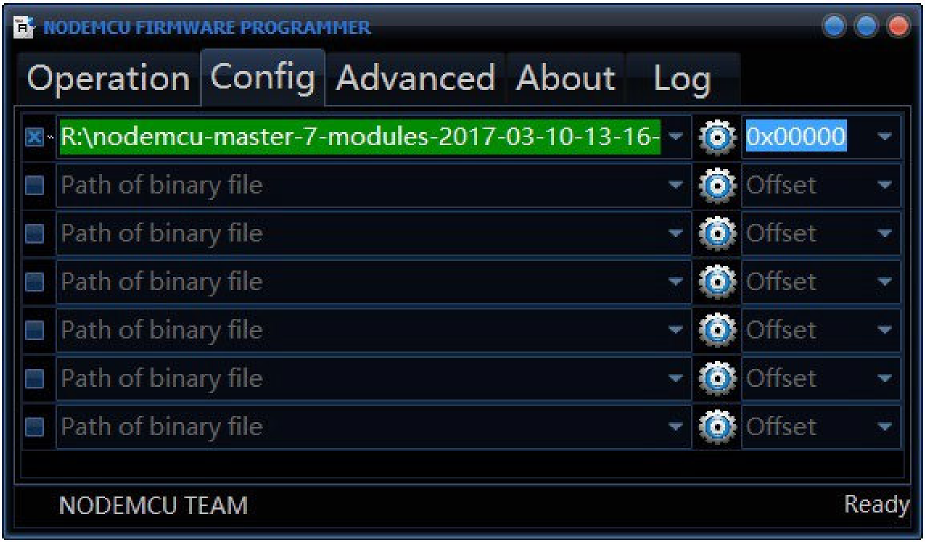
☐ ADC
☐ ADXL345
☐ AM2320
☐ APA102
☐ bit
☐ BME280
☐ BMP085
☐ CJSON
☐ CoAP
☐ Cron
☐ crypto
☐ DHT
☐ encoder
☐ end user setup
☑ file
☐ gdbstub
☑ GPIO
☐ HMC5883L
☐ HTTP
☐ HX711
☐ I^2C
☐ L3G4200D
☐ mDNS
☐ MQTT
☑ net
☑ node
☐ 1-Wire
☐ PCM
☐ perf
☐ PWM
☐ RC (no docs)
☐ rfswitch
☐ rotary
☐ RTC fifo
☐ RTC mem
☐ RTC time
☐ Sigma-delta
☐ SNTP
☐ Somfy
☐ SPI
☐ struct
☐ Switec
☐ TM1829
☑ timer
☐ TSL2561
☐ U8G
☑ UART
☐ UCG
☐ websocket
☑ WiFi
☐ WPS
☐ WS2801
☐ WS2812

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**.

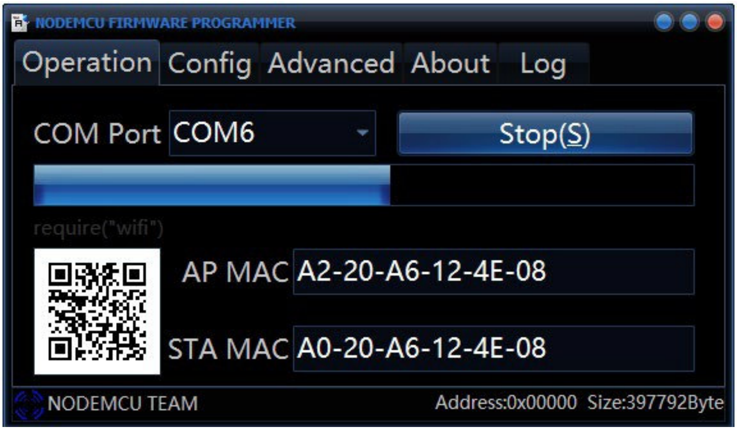


NODEMCU FIRMWARE PROGRAMMER
Operation
Config
Advanced
About
Log

☒ 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

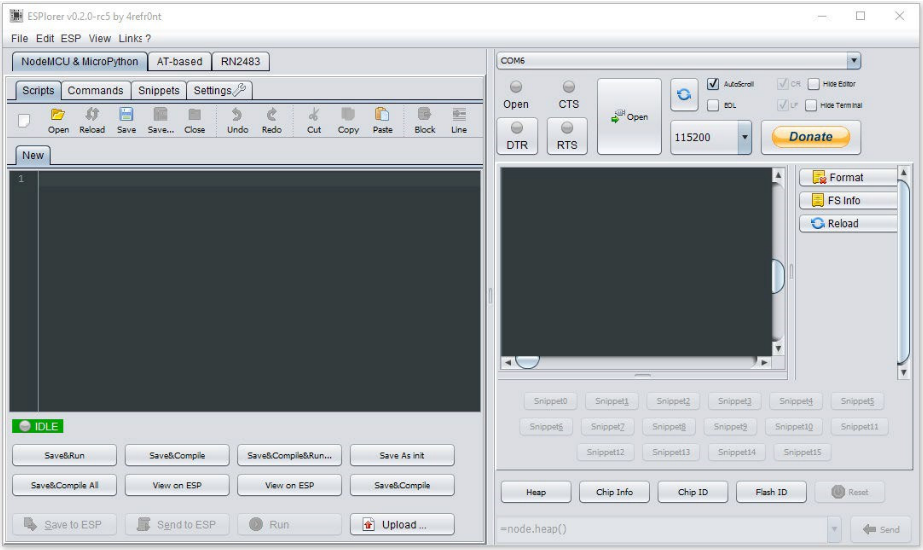
Address:0x00000 Size:397792Byte

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.

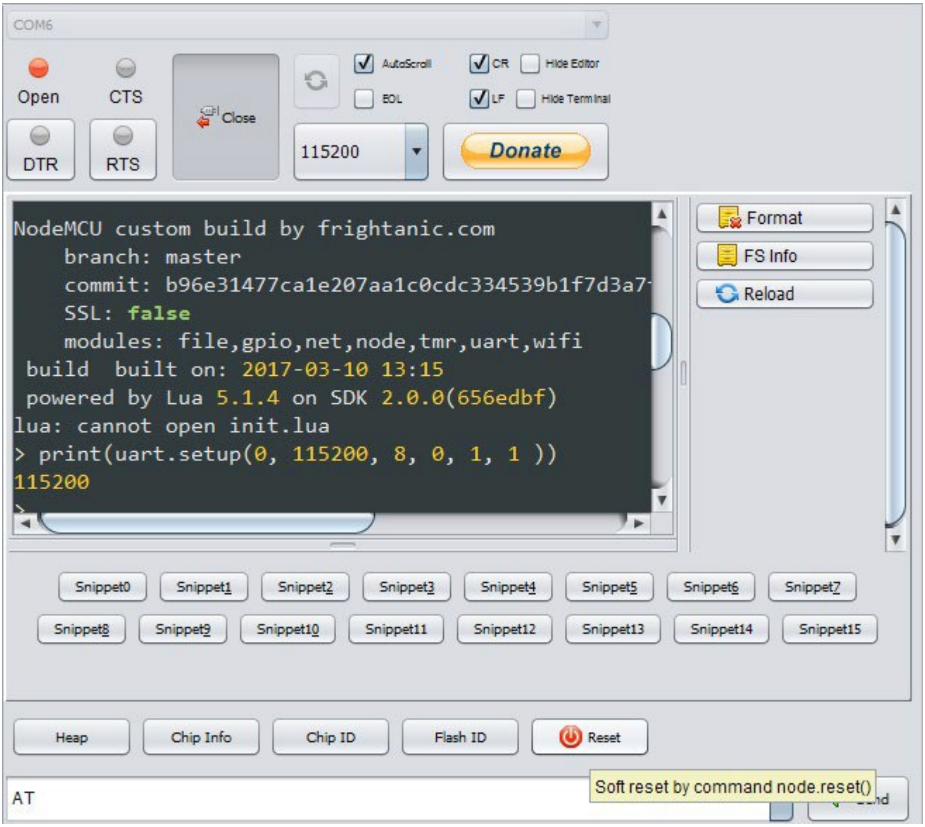
The first LUA script

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:



COM6

☑ AutoScroll ☑ CR □ Hide Editor

□ EOL ☑ LF □ Hide Terminal

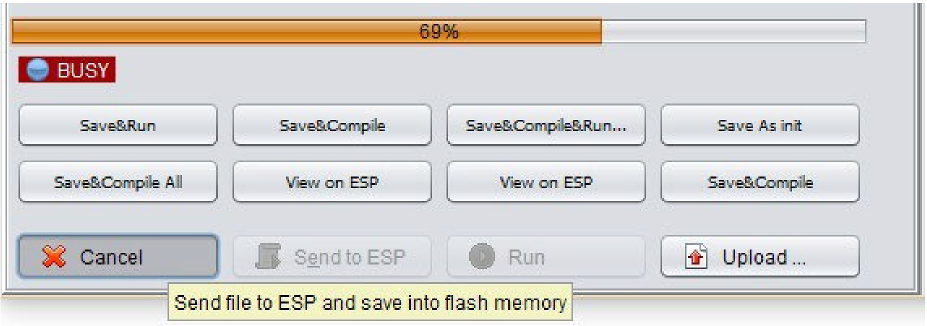
115200

```
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
>
```

AT

Soft reset by command node.reset()

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.



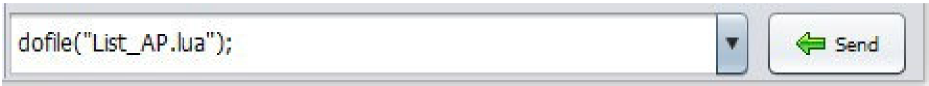
Send file to ESP and save into flash memory

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.

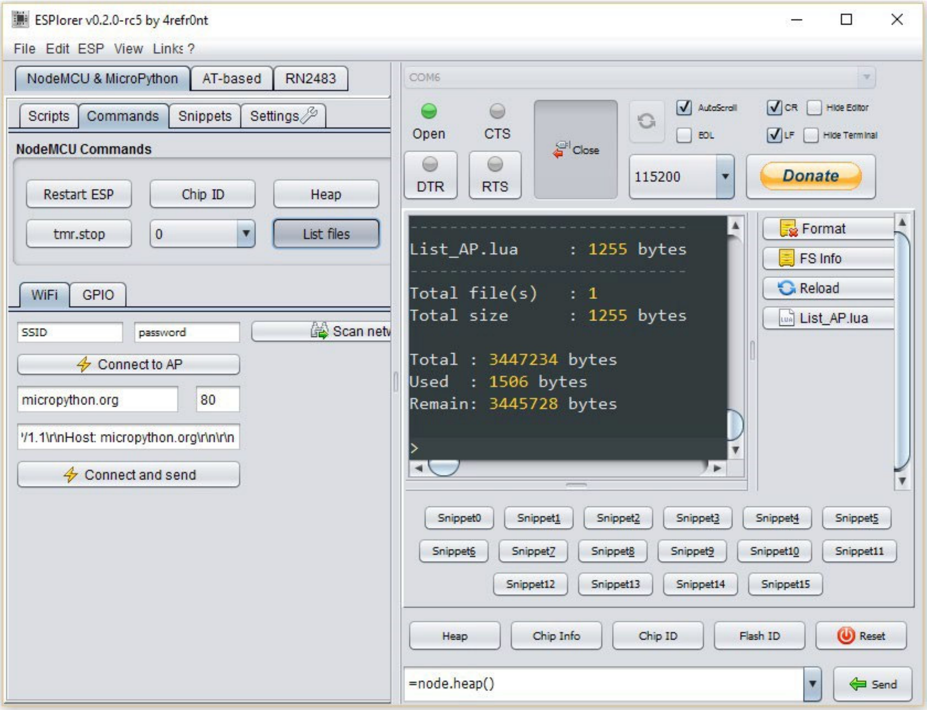
To check if the file is now on the board, go to the "**Commands**" area and click on "**List files**". The "**List_AP.lua**" file should now be listed in the terminal.

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");
```



COM6

☑ AutoScroll ☐ EOL ☑ CR ☑ LF ☐ Hide Editor ☐ Hide Terminal

115200

0

SSID password

micropython.org 80

'/1.1\r\nHost: micropython.org\r\n\r\n

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

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

>
```

=node.heap()

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