AZ-Delivery
Ihr Experte für Mikroelektronik!

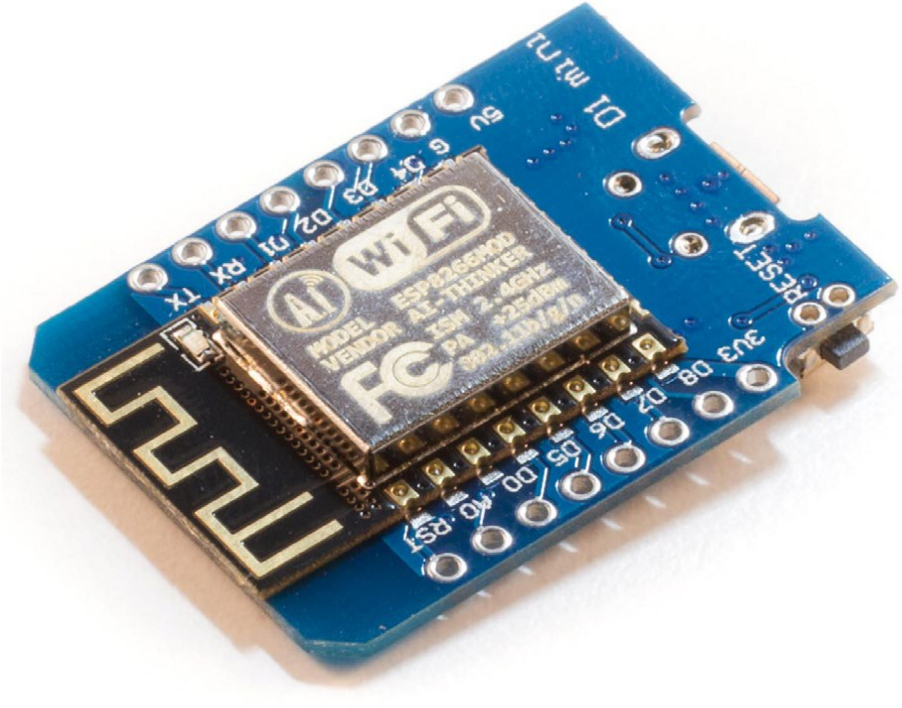
Welcome!

And thank you for purchasing our **AZ-Delivery D1 Mini NodeMCU**! In the following pages, we would go together with you from the first steps of the installation process until the first scripts.

We wish you a lot of fun!

http://flyt.it/D1Mini

The Pin-Layout of the **AZ-Delivery D1 Mini NodeMCU** corresponds to the 2.3 version of the original from the WeMos



Electronics Company. It possesses a Breadboard-friendly width, such as the width of NodeMCU V2 Amica, and thanks to the **CH340G** USB-converter, the same system compatibility as the NodeMCU Lolin V3. Moreover, it is considerably smaller than the other two, and in addition, has fewer executed pins. It is operated by a Micro-USB-port.

Overview of the most important information

» Programming via Micro USB-B cable

» Powered by:
  » Micro USB-B on the USB port of the computer
  » Micro USB-B on the 5V USB power supply unit

» 9 digital I / O pins (3,3V!)
» 1 analog I / O pin
» ESP8266 controller, variant ESP-12F
» CH340G USB interface
» Porgrammable via Arduino 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 Arduino Code,
followed by
» system preparation enabling work with Lua
and instructions for
» the first Lua script.

Overview of all Links

WeMos D1 Mini circuit diagram:

» *https://wiki.wemos.cc/_media/products:d1:mini_new_v2_2_0.pdf*

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/nodem- cu-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/mas- ter/list_ap.lua*

Other Tools:

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

Additional Interesting Information and Materials from AZ-Delivery

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

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

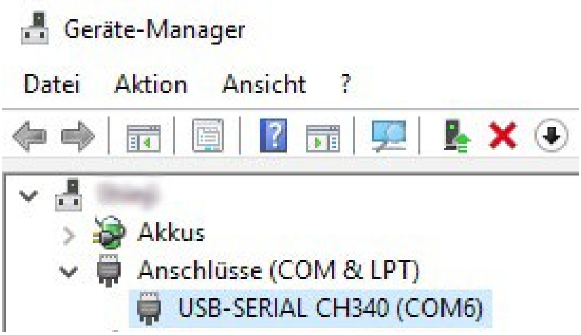
Installation of the driver

You can connect the **AZ-Delivery D1 Mini NodeMCU** to your computer via a micro USB cable. As with most of the AZ-Delivery boards, a **CH340-Chip** is used for communication, which is automatically detected and recognized by Windows.

If by any chance it does not work, here you can download the latest driver and decompress it.

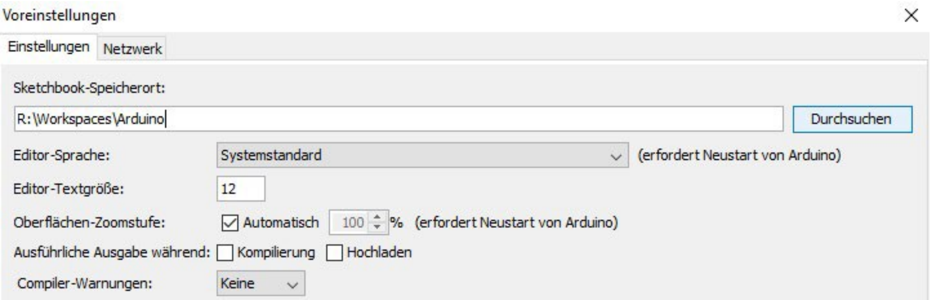
» Windows: *http://www.wch.cn/download/CH341SER_ZIP.html*
» Mac: *http://www.wch.cn/download/CH341SER_MAC_ZIP.html*

For Windows, you can simply install it by running "**SETUP.EXE**", which is located in the "**CH341SER**" folder. Mac users are best advised to follow the installation instructions that come enclosed with the driver's package.
After reconnecting the UNO, it should be recognized as "**USB-SERIAL CH340**" 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 provided there. The following first steps use desktop variations for Windows.



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

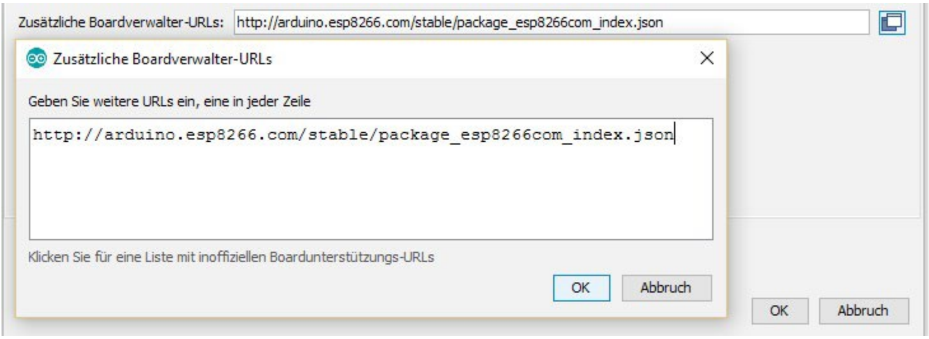
Compiler-Warnungen: Keine

If the program has been started, then the first sketchbook should have been saved under **Files > Preferences**, for example, under **My Documents\Arduino**. That way you have your, so called, Arduino "**Sketche**" scripts where you prefer and want them to be.

The NodeMCU, on the other hand, does not belong to the repertoire of the IDE, which is why the board manager needs to be expanded.

In the same window and under "**Additional Board Administrator-URLs**" add the following web address:

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



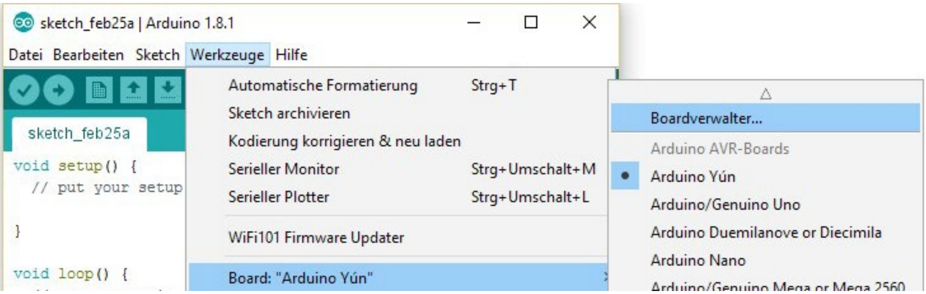
Zusätzliche Boardverwalter-URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

Zusätzliche Boardverwalter-URLs

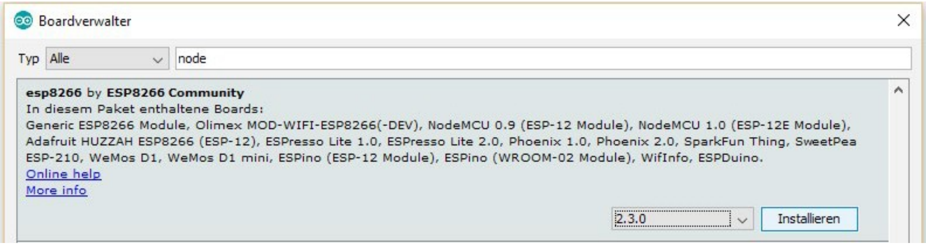
Geben Sie weitere URLs ein, eine in jeder Zeile

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

Klicken Sie für eine Liste mit inoffiziellen Boardunterstützungs-URLs

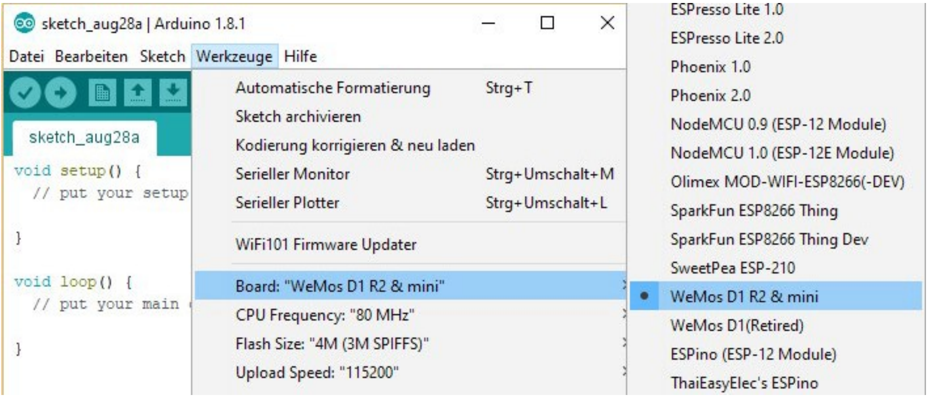

sketch_feb25a | Arduino 1.8.1
Datei Bearbeiten Sketch Werkzeuge Hilfe
Automatische Formatierung Strg+T
Sketch archivieren
Kodierung korrigieren & neu laden
Serieller Monitor Strg+Umschalt+M
Serieller Plotter Strg+Umschalt+L
WiFi101 Firmware Updater
Board: "Arduino Yún"
Boardverwalter...
Arduino AVR-Boards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

Once completed, go 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.
Online help
More info

Now you can choose the correct Board names "**WeMos D1 R2 & mini**", in addition to an **80 MHz** CPU frequency,
"**4M (3M SPIFFS)**" memory size, an upload speed of, for example, **115200** and the appropriate port ("**COM**" for Windows, "**ttyUSB**" for MacOS).

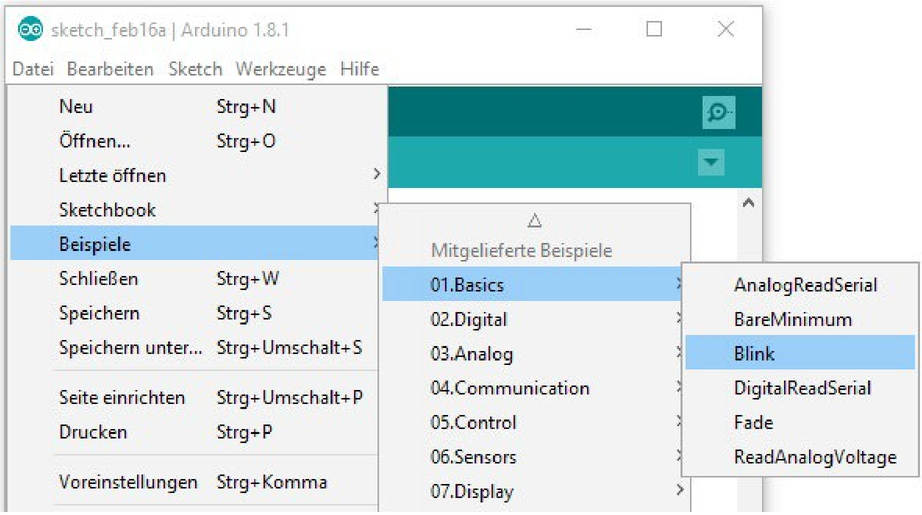

sketch_aug28a | Arduino 1.8.1
Datei Bearbeiten Sketch Werkzeuge Hilfe
Automatische Formatierung Strg+T
Sketch archivieren
Kodierung korrigieren & neu laden
Serieller Monitor Strg+Umschalt+M
Serieller Plotter Strg+Umschalt+L
WiFi101 Firmware Updater
ESPresso Lite 1.0
ESPresso Lite 2.0
Phoenix 1.0
Phoenix 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
sketch_aug28a
void setup() {
// put your setup
}
void loop() {
// put your main
}

Board: "WeMos D1 R2 & mini"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "115200"

The first script by Arduino Code

Although in most programming languages the first signal for success is the appearance of “Hello World!”, however, by Arduino is the blinking of the onboard LED. In accordance, the script is called "**Blink**".

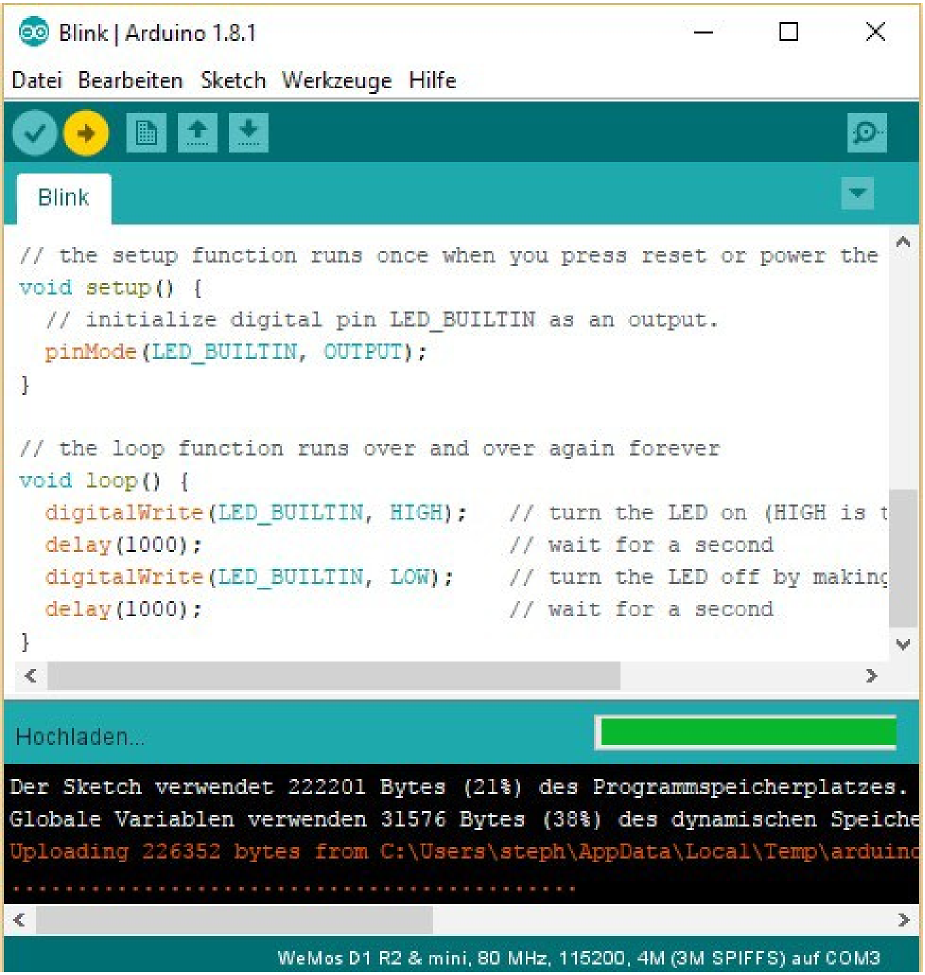
» Start the Arduino IDE and open under "**Start**" the Blink script.



Each sketch always contains the "**setup**" and "**loop**" methods. The first is performed initially and is usually used for initialization of pins and associated hardware. The loop method is then repeated continuously and thus contains most functions.

The Board’s internal LED has been already for some time automatically selected via the IDE’s own variable "**LED_BUILTIN**". **D1 Mini** is defined for **Pin D4**. In order to address the LED, you can either leave the code as it is or alter it from "**LED_BUILTIN**" to "**D4**".

You can load the sketch to the NodeMCU with the second icon located in the toolbar.


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 makinc
  delay(1000);                       // wait for a second
}
```

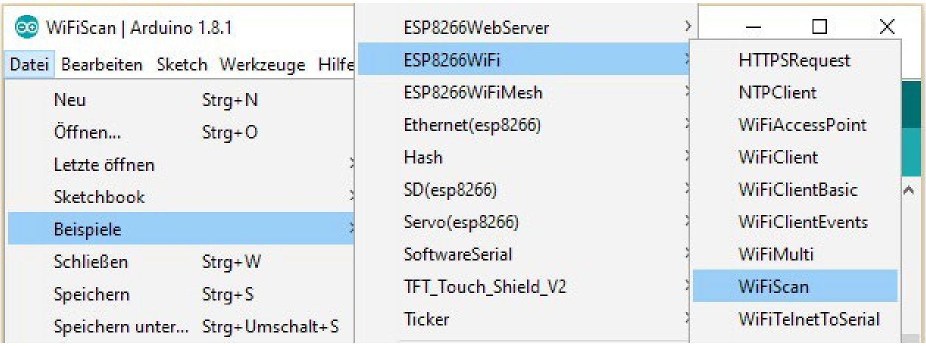
```
Der Sketch verwendet 222201 Bytes (21%) des Programmspeicherplatzes.
Globale Variablen verwenden 31576 Bytes (38%) des dynamischen Speiche
Uploading 226352 bytes from C:\Users\steph\AppData\Local\Temp\arduinc
.........................................
```

If the upload was successful, then the LED of you D1 Mini will be flashing every second.

You did it! Congratulations!

Next, you should try NodeMCU's special feature, specifically the WLAN module.

Load the "**WiFiScan**" sketch on your board and then start the serial monitor with the correct Braud rate. In a few seconds, you should see a list of all transmitting WLAN access points located in your area with the respective signal strength.



With the help of an Arduino Code, you can achieve so much more with one NodeMCU. For more options and additional examples of sketches, commence your search in the Arduino library and on the web, for example, *http://michael-sarduino.blogspot.de/search?q=8266*. Our online shop also provides hardware support:

https://az-delivery.de

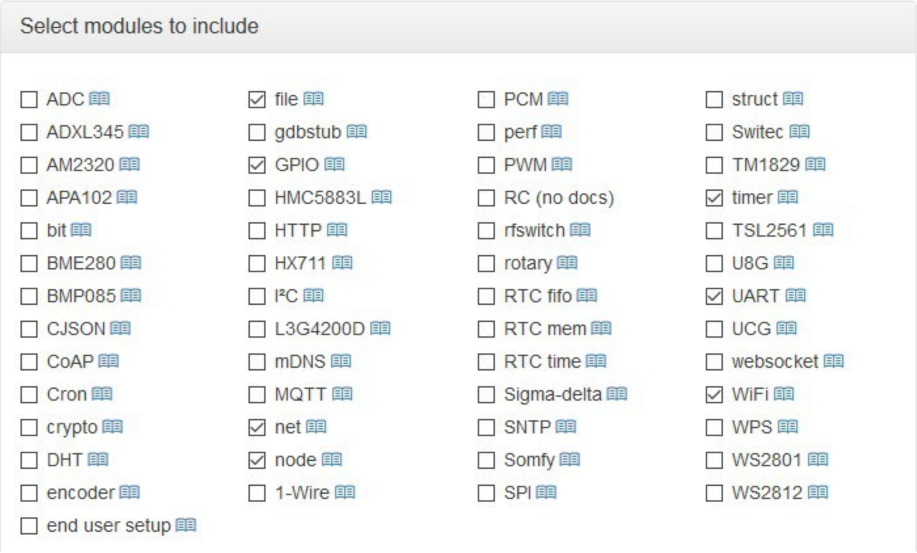
If you, however, want to immediately go ahead and learn how to use the **D1 Mini NodeMCU** with Lua scripts, then please continue to the following pages.

System preparation for working with LUA

The **D1 Mini** normally comes with an AT-Firmware from the manufacturer's AI-Thinker. In order for the LUA script language to be used with the chipset, the basis for this must be first created. In order to do that, you need to put together the right firmware specifically for your project:

» *https://nodemcu-build.com/*

Next, to the selection of the stable or developer's version, plenty of options are available, which allow you to extend the functionality of your board. However, selecting too many extensions will only slow the NodeMCU down. For our tutorial script, the standard specifications will be enough.



Select modules to include

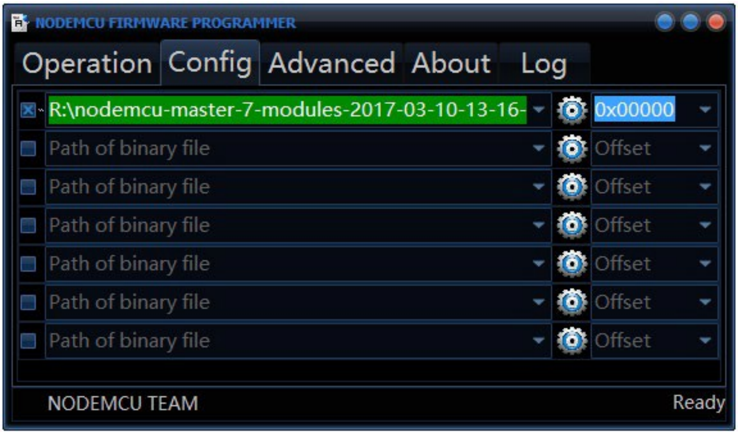
☐ ADC
☐ ADXL345
☐ AM2320
☐ APA102
☐ bit
☐ BME280
☐ BMP085
☐ CJSON
☐ CoAP
☐ Cron
☐ crypto
☐ DHT
☐ encoder
☐ end user setup
☑ file
☐ gdbstub
☑ GPIO
☐ HMC5883L
☐ HTTP
☐ HX711
☐ I^2C
☐ L3G4200D
☐ mDNS
☐ MQTT
☑ net
☑ node
☐ 1-Wire
☐ PCM
☐ perf
☐ PWM
☐ RC (no docs)
☐ rfswitch
☐ rotary
☐ RTC fifo
☐ RTC mem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

Provide your e-mail address two times in the first block and then click "**Start your build**", which is located at the bottom. In the next few minutes, you will receive an order confirmation and an e-mail with the links, from where you can download the firmware. There would be an integer and a float version from which you can choose. The only difference is that the latter can handle floating point numbers. For our tutorial, it is irrelevant which variant you choose.

In order to install the firmware, a Flash tool is necessary, as the system's independent Python script "**esptool.py**". With Windows you can comfortably use "**NodeMCU-Flasher**"program, which you can download 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

☒ 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

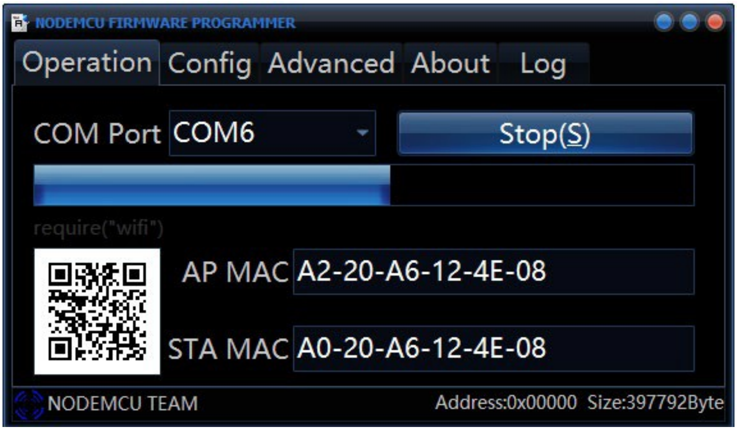
□ Path of binary file Offset

□ Path of binary file Offset

□ Path of binary file Offset

NODEMCU TEAM Ready

Under "**Advanced**" you will find options for fine adjustments to the board. For us the default settings are satisfactory, with a Braud rate of **115200**, **4 MB** flash memory, **40 MHz** memory speed and the "**DIO**"-SPI-mode. Start the Flash-Process for the COM-Port of your connected NodeMCU and wait for the green tick, which is supposed to appear the bottom left side of the window.



COM Port COM6

require("wifi")

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

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

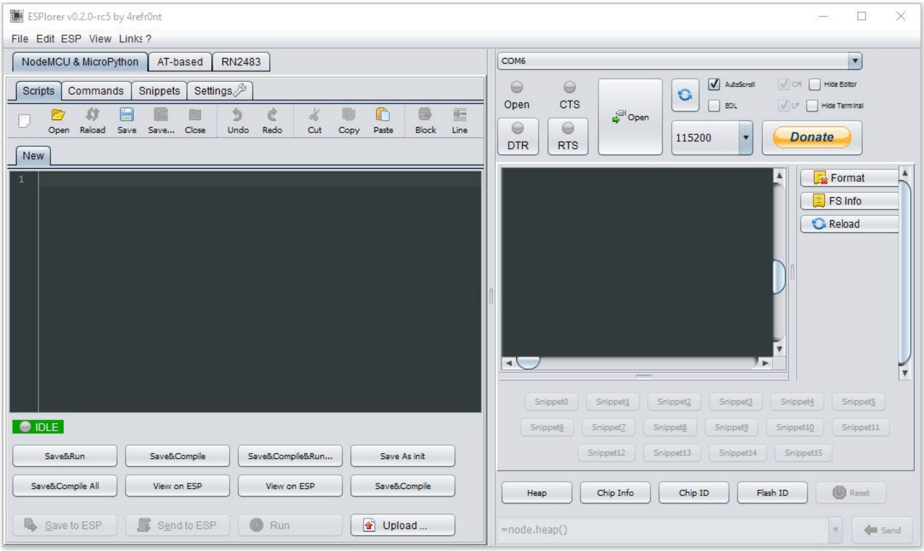
Address:0x00000 Size:397792Byte

Finally, you would need a tool that would help you write Lua scripts, and above all, load them onto the NodeMCU. The "**Luatool**" package offers pure console utilization.

The "**Esplorer**" is also platform independent and belongs with its graphical user's interface to the most popular variations. We would also use it for the tutorial:

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

Download the appropriate version for your computer system and start the "**Esplorer.bat**" (Windows) after unzipping the archive.



As you can see, the program comes with some predefined commands and, next to the NodeMCU, is able to handle other systems simultaneously. If we previously had not flashed a new firmware, now we could not have been able to use the limited commands under "**AT-based**" tab.

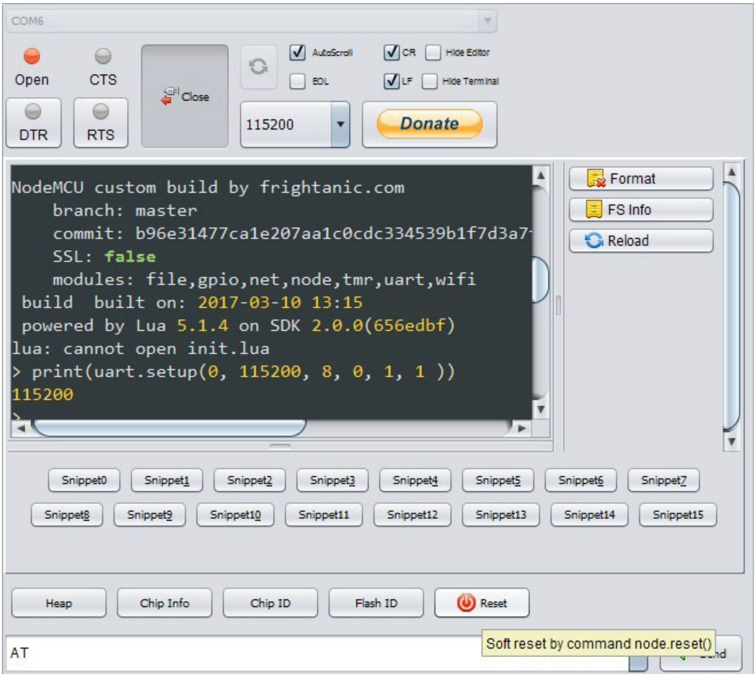
The First LUA Script

A similar function such as the "Hello World" script must come for the NodeMCU, same with the "**WiFiScan**" for the Arduino IDE. The finished code can be copied from the following webpage:

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

First, you have to check if the installation process from the previous step had worked. To do that, choose the COM-Port, located on the right side, the Baud rate (**115200** for us), and then click on "**Open**", which is again located on the right side of the window.

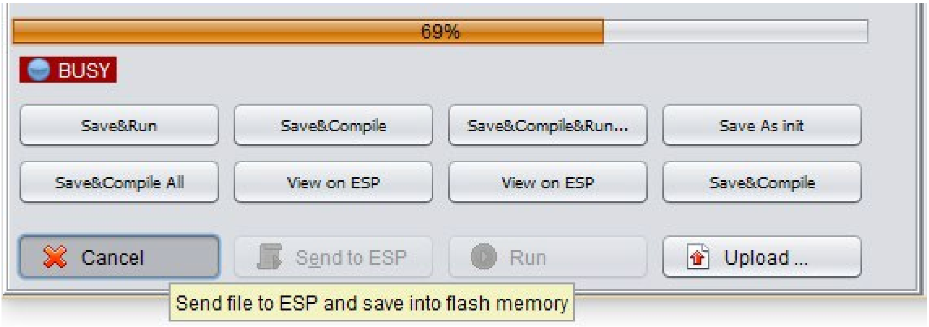
The terminal will display "**Communication with MCU..**" Now press the reset button on the NodeMCU and the board will run the boot routine indicating the information about the installed firmware. It should look similar to this screenshot:



```
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, which is located in the dark window on the left-hand side of the program page, under the "**Scripts**" tab.

If you would like to shorten the process, or are simply impatient, you can click on "**Send to ESP**", located down below. The code is then executed line by line on the NodeMCU and the result is distributed in the terminal, without having to save the script on the board.

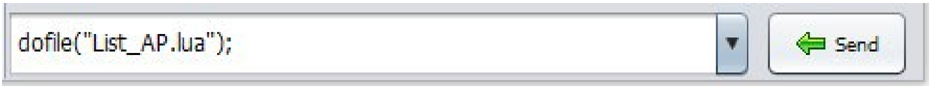


Send file to ESP and save into flash memory

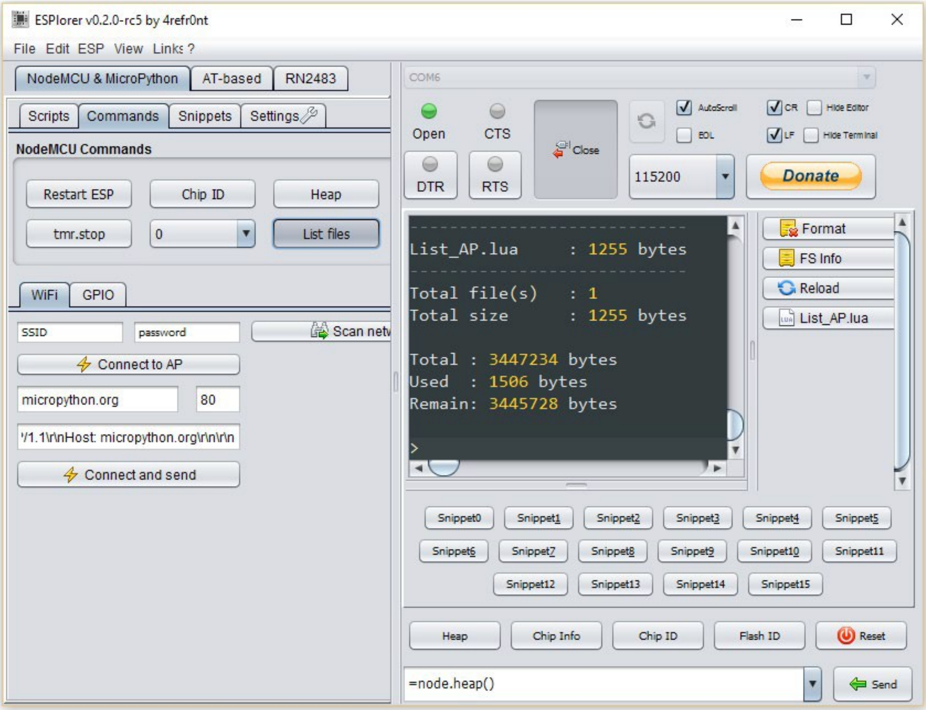
To save, click "**Save**" located above the script and give the script a name, e.g. "**List_AP.lua**". It is then automatically loaded and executed on the NodeMCU. If not, click "**Save to ESP**" located in the lower left corner.

To check whether the file is now on the board, go to the "**Commands**" area on the right, as shown in the picture, and click on "**List files**". The "**List_AP.lua**" file should now be listed in the terminal.
The script can now be also started from the command line on the bottom right with the following command directly from the D1 Mini:

```
» dofile("List_AP.lua");
```



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



COM6

☑ AutoScroll ☑ CR □ Hide Editor
□ EOL ☑ LF □ 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 online tutorials. You can start your search here: *http://nodemcu.com/index_en.html#fr_5475f7667976d8501100000f*

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

https://az-delivery.de

Enjoy!

Imprint

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