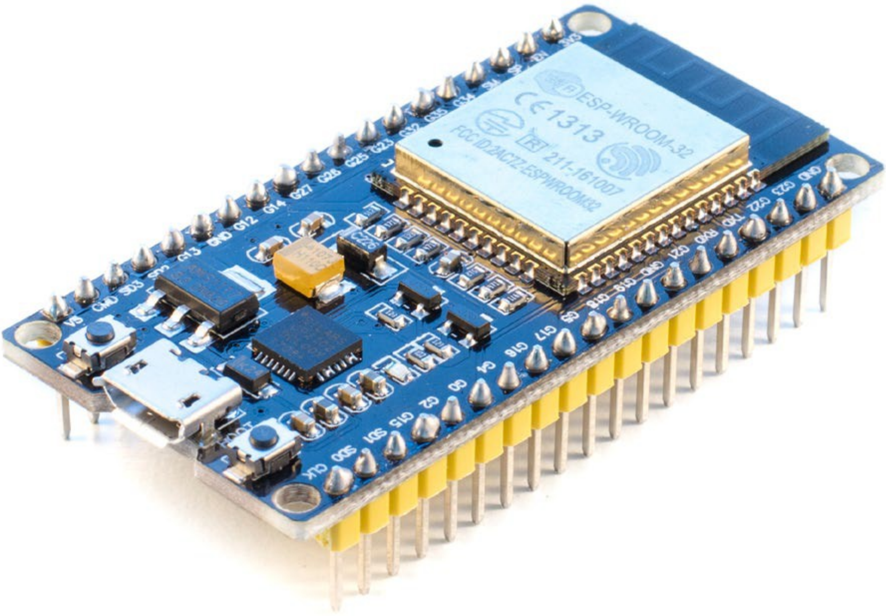
Welcome!

And thank you for purchasing our **AZ-Delivery ESP-32 Development Boards**! On the following pages, we will take you through the first steps of the installation process to the first scripts. We wish you a lot of fun!



http://flyt.it/ESP32-Devboard

The **ESP32-Chip** is a powerful successor to the very popular **ESP8266**, which, for example, is installed in the **AZ-Delivery NodeMCUs Amica V2** and **Lolin V3**. The biggest reformation of the chip is the added Bluetooth 4.2 BLE connectivity. The **AZ-Delivery ESP-32 Development Board** is the perfect companion for flexible operation in the Internet of Things.

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

» ESP-WROOM-32 processor
  » WLAN 802.11 b/g/n & Bluetooth 4.2 / BLE
  » 160MHz Tensilica L108 32 bit Dual-Core CPU
  » 512 KB SRAM & 16 MB flash memory
» 32 digital I / O pins (3,3V!)
» 6 analog-to-digital-pins
» 3x UART, 2x SPI, 2x I²C
» CP2102 USB-to-UART-interface
» Programmable via Arduino Code, Lua, MicroPython,...

On the following pages, you will find more information about
» *Driver installation and preparation of the Arduino IDE*,
And instructions for
» *the first script by Arduino Code*.

Overview of all Links

Driver:

» Windows / MacOSX / Linux/ Android:
http://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

Arduino-/Lua-/MicroPython-Services:

» Arduino IDE: *https://www.arduino.cc/en/Main/Software*
» Arduino Core: *https://github.com/espressif/arduino-esp32*
» LuaNode for ESP: *https://github.com/Nicholas3388/LuaNode*
» MicroPython Firmware for ESP32-Boards:
https://micropython.org/download/#esp32
» Espressif Flash Download Tools (Windows): *https://espressif.com/en/products/hardware/esp32/resources* (Tools)
» Esplorer: *http://esp8266.ru/esplorer/*

Other tools:

» Python: *https://www.python.org/downloads/*
» Espressif IoT Development Framework:
https://github.com/espressif/esp-idf

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

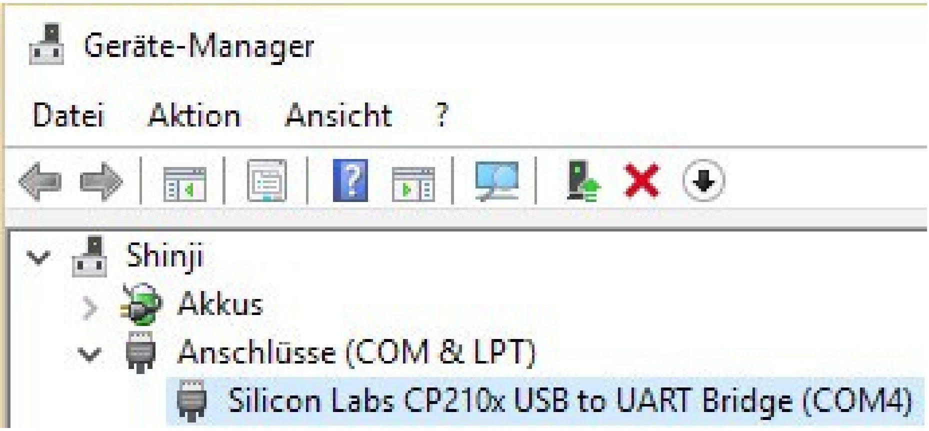
The **AZ-Delivery ESP32 Development Board** connects to your computer via a Micro-USB-cable. The microcontroller uses a **CP2102-Chip** for the USB-interface, which is normally recognized by Windows partly recognized by MacOS systems.

If that, however, is not the case, then please download the current driver from the link below, 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 in your loaded archive.

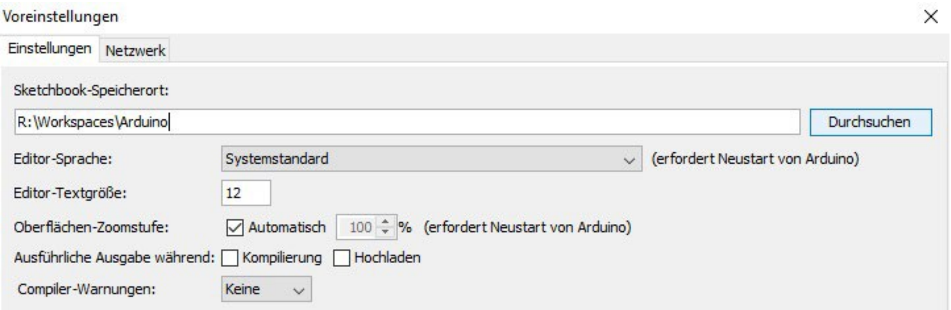
After reconnecting the NodeMCU, it should be recognized as a "**Silicon Labs CP210x USB to UART Bridge**"-device (Windows).



Shinji
Akkus
Anschlüsse (COM & LPT)
Silicon Labs CP210x USB to UART Bridge (COM4)

Installation of the ESP32 Development Boards

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 in there. The following first steps use the Windows desktop version.



Voreinstellungen

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

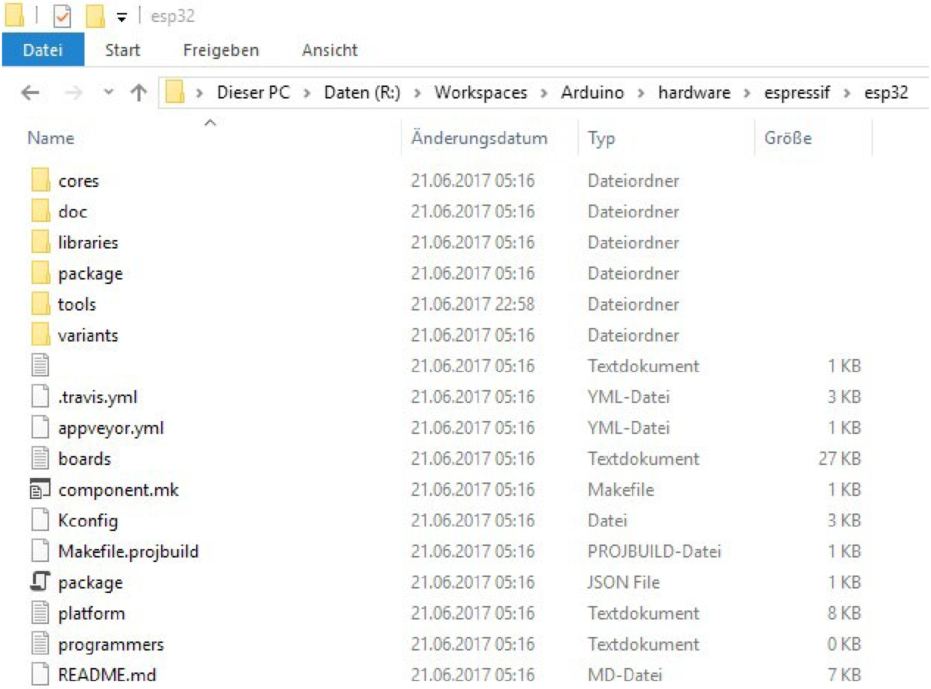
Compiler-Warnungen: Keine

If the program has been started, then the storage location of the first sketchbook, should be set under **File > Preferences,** for example under **my documents\Arduino**. That way your scripts by Arduino, which are named "**Sketche**" will be stored at the place you want and prefer.

The **ESP32** is not part of the IDE's standard repertoire, and he is not to be found in the Board-Manager. From here you can download the Arduino Core-files for the controller:

» *https://github.com/espressif/arduino-esp32/archive/master.zip*

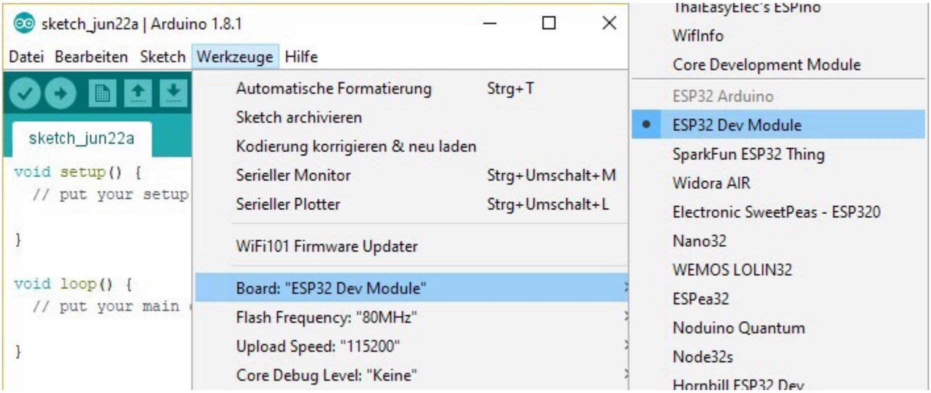
Once you have completed that, unzip the contents of the **"arduino-esp32-master"** directory into the hardware folder of your Arduino-Sketchbook under (e.g. my documents\...) **\Arduino\hardware\espressif\esp32\**. Then start the **"get.exe"** application, which is located in the **tools**-directory, and wait for the complete download of the required files.



Dieser PC > Daten (R:) > Workspaces > Arduino > hardware > espressif > esp32

Name	Änderungsdatum	Typ	Größe
cores	21.06.2017 05:16	Dateiordner	
doc	21.06.2017 05:16	Dateiordner	
libraries	21.06.2017 05:16	Dateiordner	
package	21.06.2017 05:16	Dateiordner	
tools	21.06.2017 22:58	Dateiordner	
variants	21.06.2017 05:16	Dateiordner	
	21.06.2017 05:16	Textdokument	1 KB
.travis.yml	21.06.2017 05:16	YML-Datei	3 KB
appveyor.yml	21.06.2017 05:16	YML-Datei	1 KB
boards	21.06.2017 05:16	Textdokument	27 KB
component.mk	21.06.2017 05:16	Makefile	1 KB
Kconfig	21.06.2017 05:16	Datei	3 KB
Makefile.projbuild	21.06.2017 05:16	PROJBUILD-Datei	1 KB
package	21.06.2017 05:16	JSON File	1 KB
platform	21.06.2017 05:16	Textdokument	8 KB
programmers	21.06.2017 05:16	Textdokument	0 KB
README.md	21.06.2017 05:16	MD-Datei	7 KB

Then, when you open the Arduino IDE you can select under **"Tools > Board"** the **"ESP32 Dev Module"**, and, in addition, a flash frequency of **80MHz** and a baud rate of at least **"115200"**.



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

}

void loop() {
  // put your main

}
```

Board: "ESP32 Dev Module"

Flash Frequency: "80MHz"

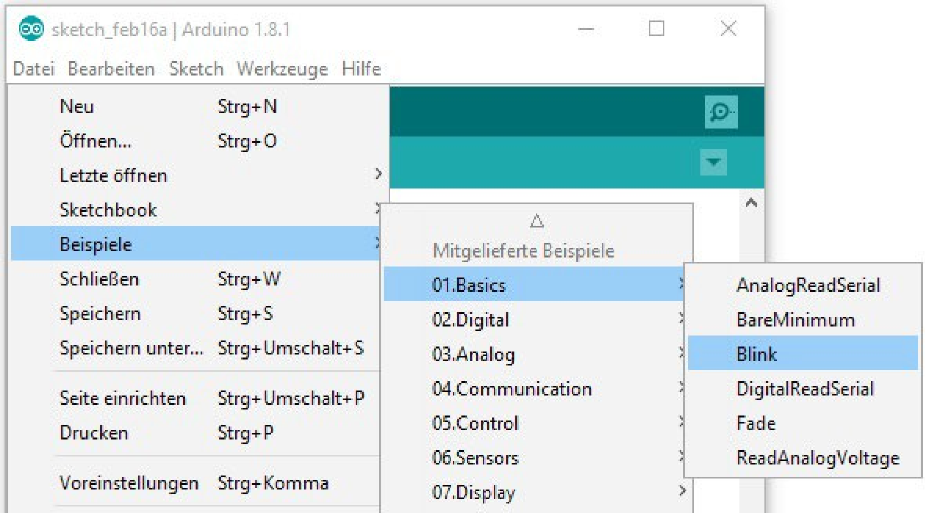
Upload Speed: "115200"

Core Debug Level: "Keine"

The First Script

Although, in most programming languages the first sign of success is the appearance of the "Hello World!" phrase, when it comes to Arduino and other microcontrollers, however, the first sign of success is the blinking of the internal board's LED. In accordance, the script is called "**Blink**".

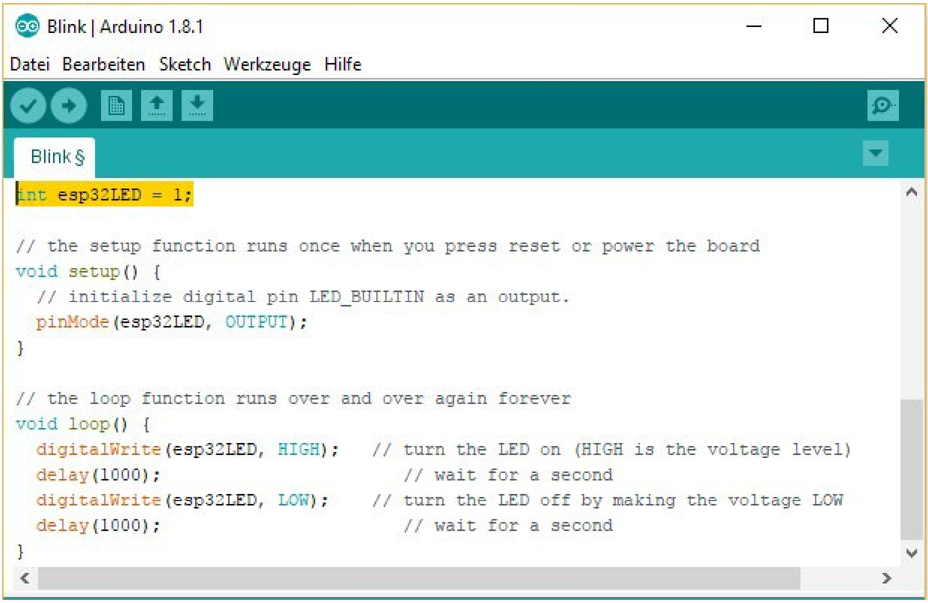
» Start the Arduino IDE and open the Blink-Script under "**Start**".



The methods "**setup**" and "**loop**" are always contained in each sketch. The former is always initially executed and is normally used to initialize pins and separated hardware. After that, the loop method is executed and permanently repeated. For this reason, it contains almost all other functions.

The board-internal LED has automatically been selected for some time via the IDE's own variable "**LED_BUILTIN**". Since the ESP32-Core files for the Arduino IDE are still under development and the

Pin layouts vary very much depending on the manufacturer, this variable does not work here. The board's internal LED of the AZ-Delivery ESP32 Development Boards lies on Pin 1. Change the Sketch as shown in the middle picture.

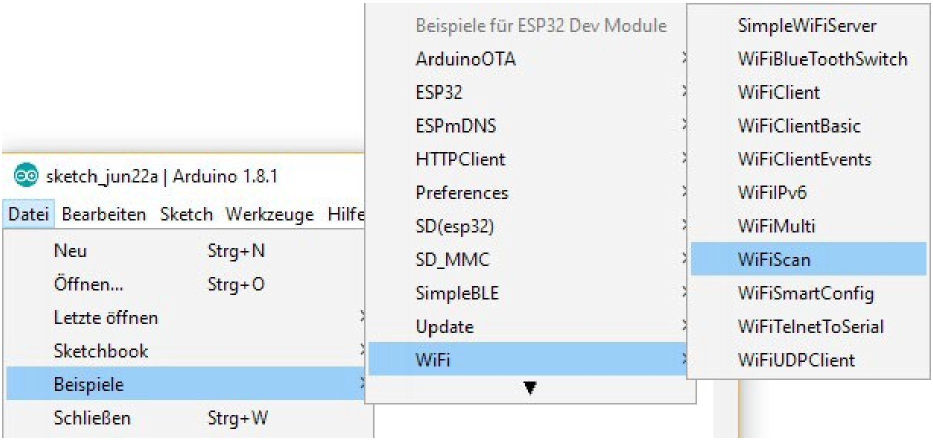


```
int esp32LED = 1;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(esp32LED, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(esp32LED, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(esp32LED, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

With the second symbol (the arrow showing to the right) under the command bar, you can load the sketch onto the NodeMCU.
If the upload was successful, then the LED of your board will start blinking every second.

You did it! Congratulations!

Next, you should look at the existing scripts for the ESP32, for example, "WifiScan". Compare the baud rates of the code with your settings. Upload the sketch to the Development Board. Finally, after a few seconds, you should see a list of all available WLAN-Access-Points that are in your surroundings, as well as their respective signal strength.



Thanks to the Arduino Code you can achieve so much more with the ESP32 Development Board. Start your search for further possibilities, best from the multiple examples of sketches in the Arduino library and in the web, here at:

http://michaelsarduino.blogspot.de/search?q=8266.

For hardware support, our online store is always at your disposal:

https://az-delivery.de

Imprint

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