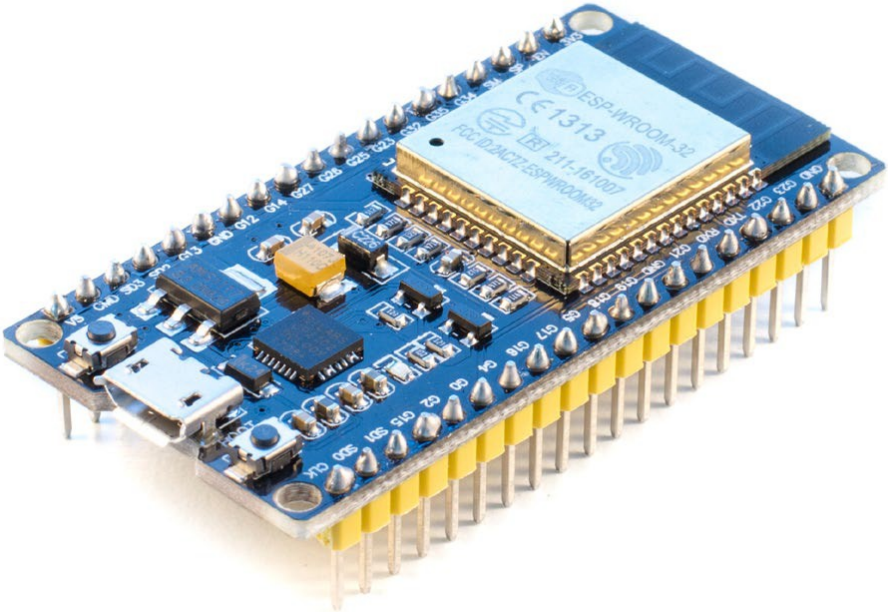
Willkommen!

Und herzlichen Dank für den Kauf unseres **AZ-Delivery ESP-32 Development-Boards**! Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Schritte von der Einrichtung bis zu den ersten Scripten. Viel Spaß!



http://flyt.it/ESP32-Devboard

Der **ESP32-Chip** ist ein leistungsstarker Nachfolger des sehr beliebten **ESP8266**, welcher beispielsweise in den **AZ-Delivery NodeMCUs Amica V2** und **Lolin V3** verbaut ist. Die größte

Neuerung ist dabei die hinzugekommene Bluetooth 4.2 BLE Konnektivität. Das **AZ-Delivery ESP-32 Development-Board** ist der perfekte Begleiter für den flexiblen Einsatz im Internet of Things.

Die wichtigsten Informationen in Kürze

» Programmierung über Micro USB-B-Kabel

» Stromversorgung über:
  » Micro USB-B am USB-Anschluss des Rechners
  » Micro USB-B am 5V USB-Netzteil

» ESP-WROOM-32 Prozessor
  » WLAN 802.11 b/g/n & Bluetooth 4.2 / BLE
  » 160MHz Tensilica L108 32 bit Dual-Core CPU
  » 512 KB SRAM & 16 MB Flashspeicher
» 32 digitale I / O-Pins (3,3V!)
» 6 Analog-zu-Digital-Pins
» 3x UART, 2x SPI, 2x I^2C
» CP2102 USB-zu-UART-Schnittstelle
» Programmierbar über Arduino Code, Lua, MicroPython,...

Auf den nächsten Seiten findest du Informationen zur
» *Treiber-Installation und Vorbereitung der Arduino IDE*,
und eine Anleitung für
» *das erste Script per Arduino Code*.

Alle Links im Überblick

Treiber:

» 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 für ESP: *https://github.com/Nicholas3388/LuaNode*

» MicroPython Firmware für 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/*

Sonstige Tools:

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

» Espressif IoT Development Framework: *https://github.com/espressif/esp-idf*

Interessantes von AZ-Delivery

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

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

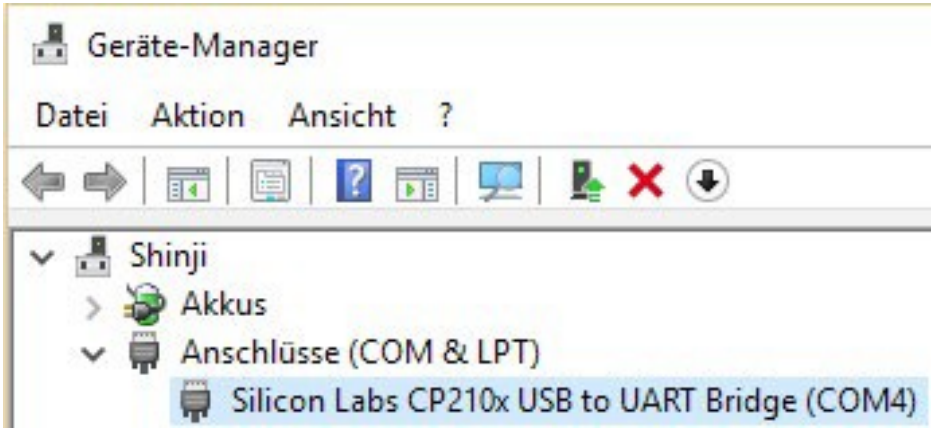
Treiberinstallation

Das **AZ-Delivery ESP32 Development Board** verbindest du über ein Micro-USB-Kabel mit deinem Rechner. Der Microcontroller nutzt für die USB-Schnittstelle einen **CP2102-Chip**, der von Windows in der Regel, bei MacOS-Systemen teilweise automatisch erkannt wird.

Sollte das nicht der Fall sein, lade dir hier den aktuellen Treiber herunter und entpacke ihn.

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

Unter Windows installierst du ihn einfach durch das Ausführen der **"CP210xVCPInstaller_x86.exe"** oder **"CP210xVCPInstaller_x64.exe"** je nach dem eigenen System. Als Mac-Nutzer installierst du die in deinem geladenen Archiv liegende DMG-Datei.
Nach dem erneuten Anschließen des NodeMCUs sollte dieser als "**Silicon Labs CP210x USB to UART Bridge**"-Gerät (Windows) erkannt werden.

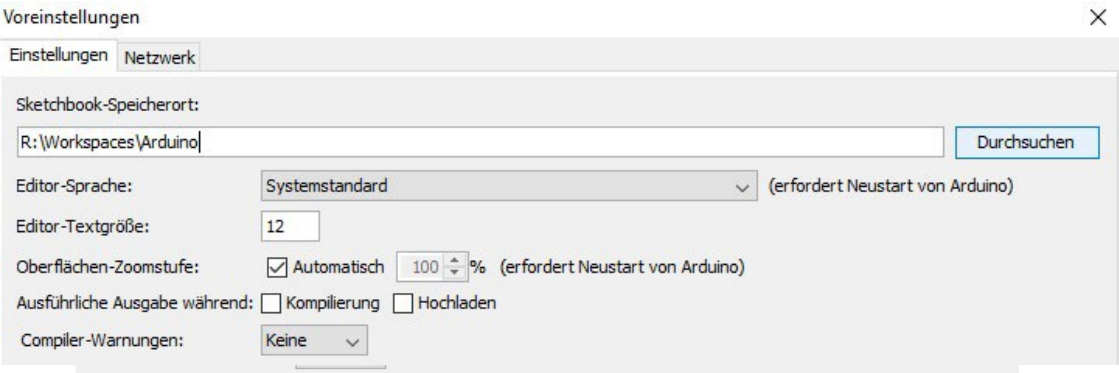


Installation des ESP32 Development Boards

Hinweis: Es gibt eine neue Methode um die benötigten Dateien automatisch über den Boardmanager zu installieren. Eine Detaillierte Schritt-für-Schritt Anleitung dazu finden Sie in unserem Blogartikel „*ESP32 jetzt über den Boardverwalter installieren*“.

Damit entfällt die in diesem Abschnitt gezeigte manuelle Installation der Dateien.

Besuche die Seite *https://www.arduino.cc/en/Main/Software* und lade die aktuelle Version für dein Betriebssystem herunter. Alternativ kannst du dich für den Arduino Web-Editor registrieren und den leicht verständlichen Installationshinweisen folgen. Die folgenden ersten Schritte nutzen Desktop-Variante für Windows.



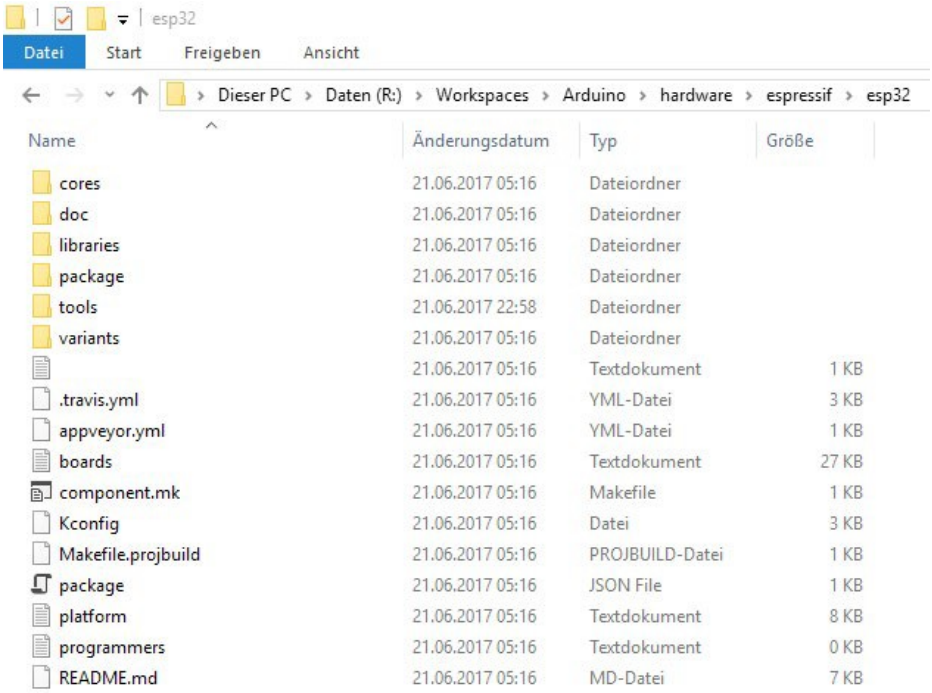
Ist das Programm gestartet, sollte unter ***Datei*** > ***Voreinstellungen*** der an erster Stelle stehende Sketchbook-Speicherort festgelegt werden, beispielsweise unter ***Eigene Dokumente\Arduino***. Damit landen deine bei Arduino "***Sketche***" genannten Scripte auch dort, wo du sie haben möchtest.

Der **ESP32** gehört allerdings nicht zum Standardrepertoire der IDE und selbst im komfortablen Board-Manager ist er noch nicht

zu finden. Lade dir daher die Arduino Core-Dateien für den Controller herunter:

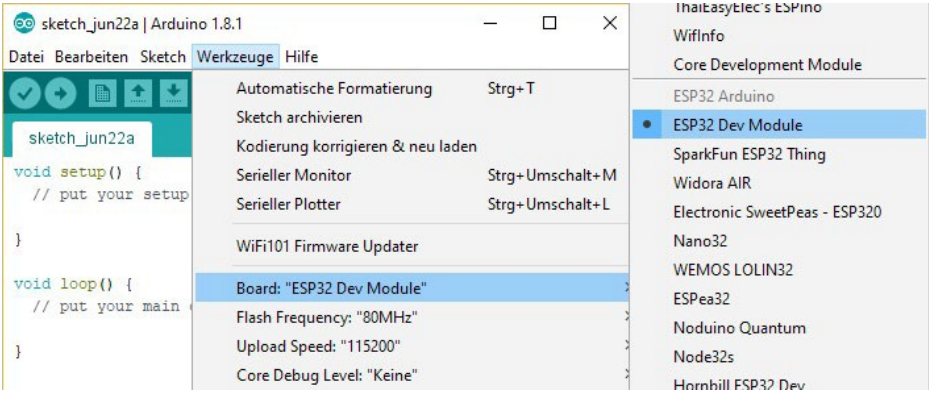
» *https://github.com/espressif/arduino-esp32/archive/master.zip*

Ist das geschafft, entpacke den Inhalt aus dem Verzeichnis **"arduino-esp32-master"** in den hardware-Ordner deines Arduino-Sketchbooks unter (z. B. Eigene Dokumente\...) **\ Arduino\ hardware\espressif\esp32**. Starte anschließend im **tools**-Verzeichnis die Anwendung **"get.exe"** und warte den vollständigen Download der benötigten Dateien ab.



Im Anschluss öffnest du die Arduino IDE und kannst unter **"Werkzeuge > Board"** das **"ESP32 Dev Module"** auswählen, dazu die Flash-Frequenz von **80MHz** und eine Baud-Rate von

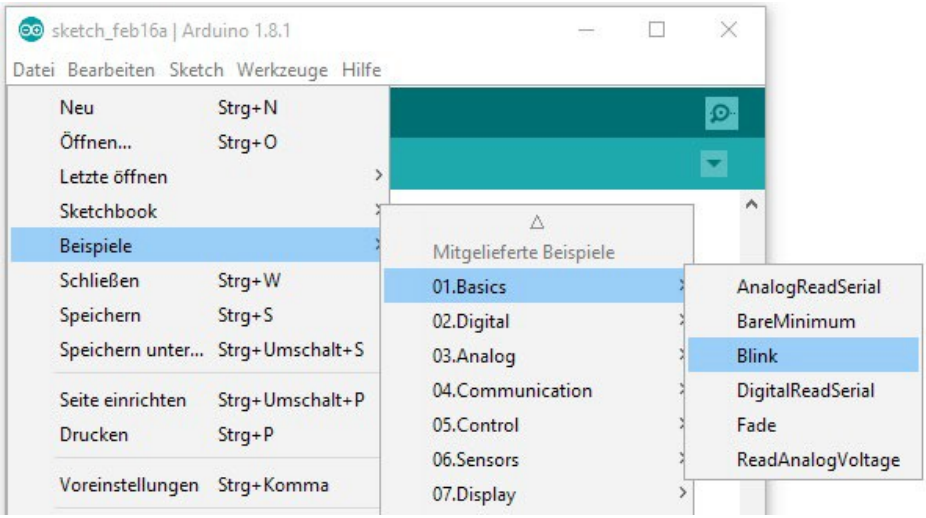
mindestens "**115200**".



Das erste Script

Während in den meisten Programmiersprachen der erste Erfolg ein zu lesendes "Hello World!" darstellt, ist es bei Arduinos und anderen Microcontrollern das Blinken der boardinternen LED. Das Script heißt entsprechend "***Blink***".

» Starte die Arduino IDE und öffne unter "**Start**" das Blink-

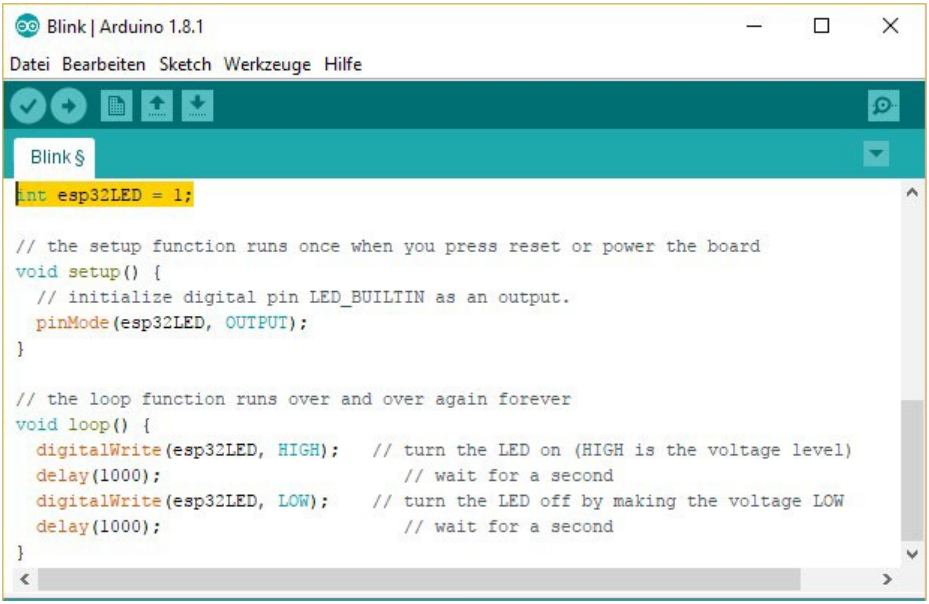


Script.

Jeder Sketch enthält immer die Methoden "***setup***" und "***loop***". Erstere wird zu Beginn ausgeführt und in der Regel zur Initialisierung von Pins und angeschlossener Hardware verwendet. Die loop-Methode wird im Anschluss permanent wiederholt und enthält damit fast alle anderen Funktionen.

Die Board-interne LED wird seit einiger Zeit über die IDE-eigene Variable "**LED_BUILTIN**" automatisch ausgewählt. Da die ESP32-

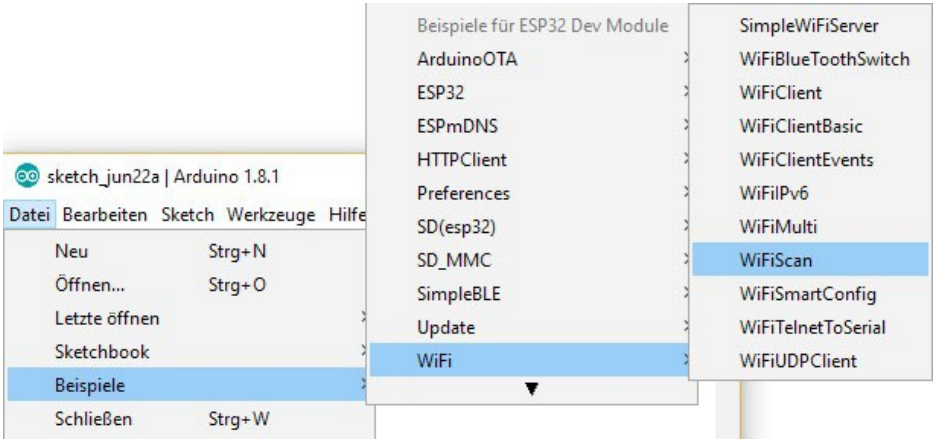
Core-Dateien für die Arduino IDE noch in der Entwicklung sind und die Pin-Layouts auch je nach Hersteller variieren, funktioniert diese Variable hier nicht. Die boardinterne LED des AZ-Delivery ESP32 Development Boards liegt an Pin 1. Ändere den Sketch wie im mittleren Bild ab.



Mit dem zweiten Symbol (Pfeil nach rechts) unter der Befehlsleiste lädst du den Sketch auf die NodeMCU.
War der Upload erfolgreich, blinkt die LED deines Boards nun im Sekundentakt.

Du hast es geschafft! Herzlichen Glückwunsch!

Als nächstes solltest du dir die vorhandenen Scripte für den ESP32 anschauen, zum Beispiel "WifiScan". Gleiche die Baudraten des Codes mit deinen Einstellungen ab lade den Sketch auf das Development Board. Im Anschluss solltest du wenige Sekunden später eine Liste aller in deiner Umgebung funkenden WLAN-Access-Points mitsamt der jeweiligen Signalstärke sehen.



Mithilfe von Arduino Code kannst du noch vieles mehr mit dem ESP32 Development Board erreichen. Starte deine Suche nach weiteren Möglichkeiten am besten bei den vielen anderen Beispielsketchen der Arduino-Bibliothek und im Web, beispielsweise auf

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

Für Hardwareunterstützung sorgt natürlich unser Online-Shop:

https://az-delivery.de

Impressum

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