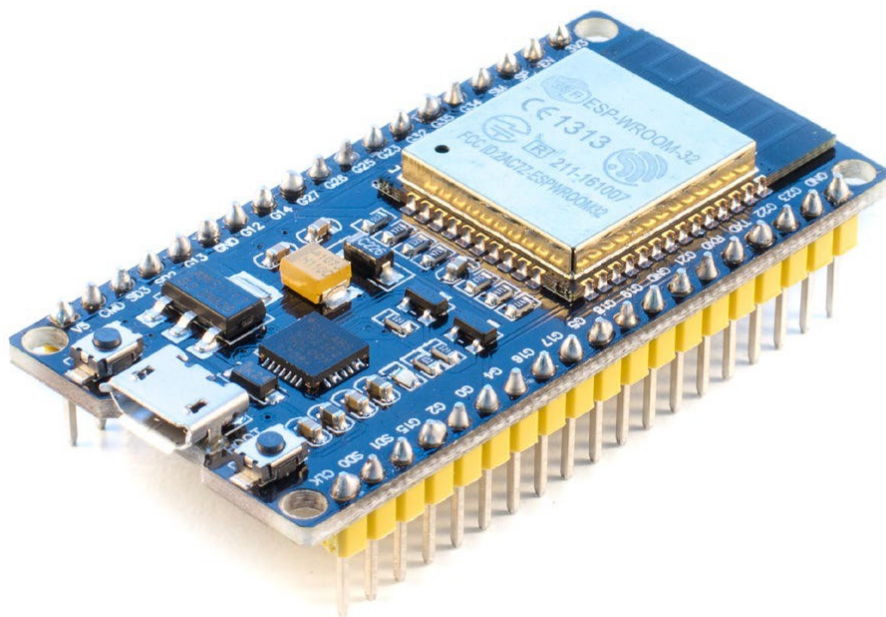


Benvenuto!

E grazie per aver acquistato **ESP-32 Development-Board di AZ-Delivery**! Nelle pagine seguenti ti accompagneremo nei primi passi dall'installazione del dispositivo fino ai primi script. Buon divertimento!



<http://flyt.it/ESP32-Devboard>

Il **chip ESP32** è un potente successore del famosissimo ESP8266, che è installato nei **NodeMCU di AZ-Delivery**, come ad esempio Amica V2 e Lolin V3. Il più grande cambiamento è l'aggiunta della connettività BLE Bluetooth 4.2. **ESP-32 Development-Board di AZ-Delivery** è il compagno perfetto per un uso flessibile nell'Internet of Things.

Le informazioni più importanti in breve

» Programmazione tramite cavo micro USB-B

- » Alimentazione elettrica tramite:
 - » Micro USB-B sulla porta USB del computer
 - » Micro USB-B sull'adattatore di alimentazione USB da 5V
- » Processore ESP-WROOM-32
 - » WLAN 802.11 b / g / n e Bluetooth 4.2 / BLE
 - » CPU dual-core a 32 bit Tensilica L108 da 160 MHz
 - » 512 KB SRAM e 16 MB di memoria flash
- » 32 pin I / O digitali (3,3 V!)
- » 6 pin analogici-digitali
- » 3x UART, 2x SPI, 2x I²C
- » Interfaccia da USB a UART CP2102
- » Programmabile tramite codice Arduino, Lua, MicroPython, ...

Nelle prossime pagine troverai informazioni su

- » *Installazione driver e preparazione dell'IDE Arduino,*
- e una guida per
- » *Il primo script di codice Arduino.*

Tutti i link utili

Driver:

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

Servizi per Arduino-/Lua-/MicroPython:

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

Altri strumenti:

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

Cose interessanti su AZ-Delivery

- » AZ-Delivery G+Community:
<https://plus.google.com/communities/115110265322509467732>
- » AZ-Delivery auf Facebook:
<https://www.facebook.com/AZDeliveryShop/>

Installazione del driver

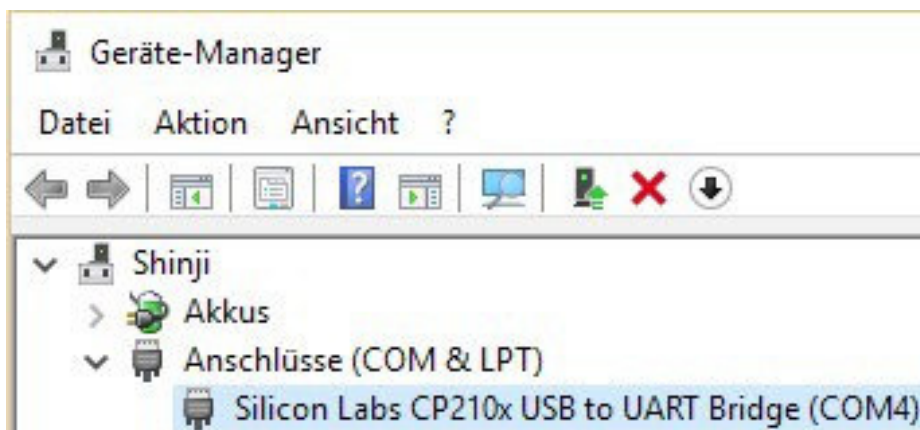
Collega la **ESP32 Development Board di AZ-Delivery** al tuo computer tramite un cavo micro USB. Il microcontrollore utilizza un **chip CP2102** per l'interfaccia USB, che viene in genere riconosciuto automaticamente da Windows e parzialmente riconosciuto dai sistemi MacOS.

Se questo non è così, scarica qui il driver corrente e decomprimilo.

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

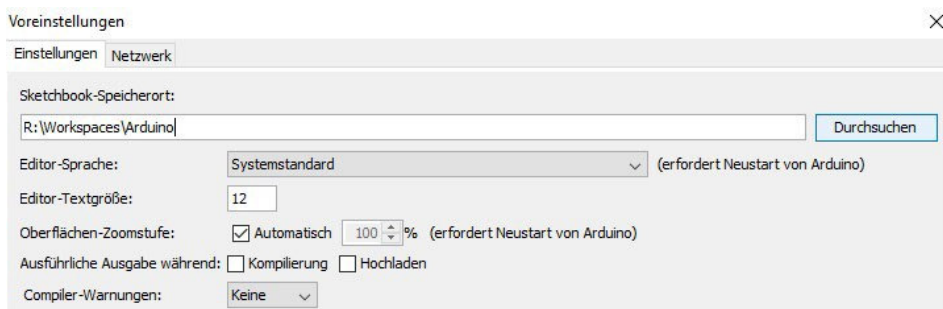
Su Windows, è sufficiente installarlo eseguendo "CP210xVCPInstaller_x86.exe" o "CP210xVCPInstaller_x64.exe" a seconda del sistema. Come utente Mac, si installa il file DMG nell'archivio caricato.

Dopo aver ricollegato il NodeMCU, dovrebbe essere riconosciuto come dispositivo "Silicon Labs CP210x da USB a UART Bridge" (Windows).



Installazione di ESP32 Development Board

Visita la pagina <https://www.arduino.cc/en/Main/Software> e scarica la versione più recente per il tuo sistema operativo. In alternativa, è possibile registrarsi per l'Arduino Web Editor e seguire le facili istruzioni di installazione. I seguenti primi passi utilizzano la variante desktop per Windows.



Quando il programma viene avviato, è necessario impostare la posizione di salvataggio di Sketchbook in **File> Preferenze**, ad esempio in **Documenti \ Arduino**. Questo farà salvare i tuoi script chiamati Arduino "**Sketches**" dove vorrai.

L'**ESP32** non fa parte del repertorio standard dell'IDE e anche nel confortevole board manager non viene ancora trovato. Scarica i file Arduino Core per il controller:

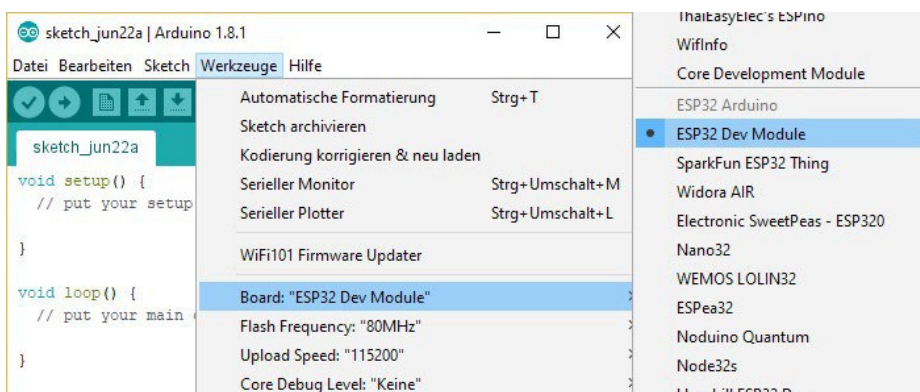
» <https://github.com/espressif/arduino-esp32/archive/master.zip>

Una volta fatto, decomprimere i contenuti dalla directory "**arduino-esp32-master**" nella cartella hardware del tuo sketchbook Arduino (ad esempio My Documents \ ...) \ **Arduino \ hardware \ espressif \ esp32** \. Quindi avviare l'applicazione "**get.exe**" nella directory degli strumenti e attendere il download completo dei file richiesti.

| 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 |

Quindi si apre l'IDE Arduino e sotto **"Strumenti> Scheda"** è

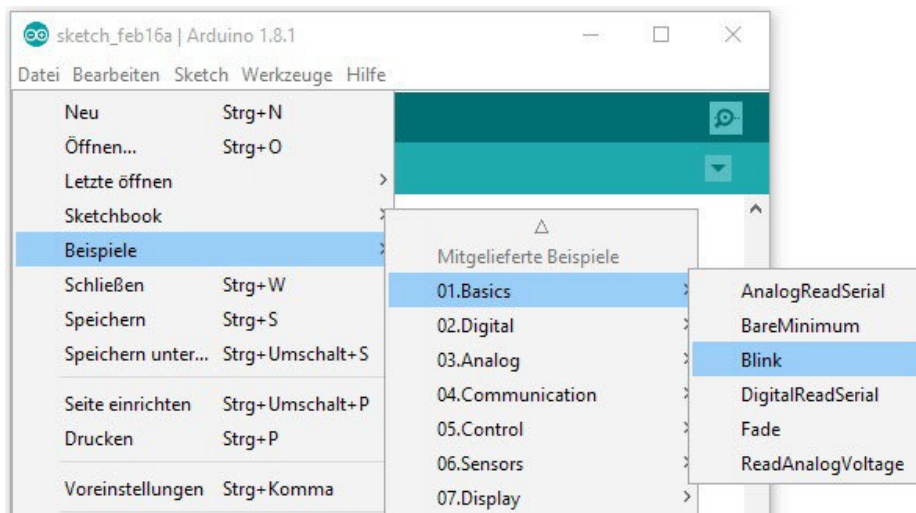
possibile selezionare **"ESP32 Dev Module"**, oltre alla frequenza flash di **80 MHz** e una velocità di trasmissione di almeno **"115200"**.



Il primo script

Mentre nella maggior parte dei linguaggi di programmazione il primo successo è un "Hello World!", in Arduino e altri microcontrollori, è il lampeggiare del LED di bordo. Lo script è chiamato **"Blink"**.

» Avvia l'IDE Arduino e apri lo script-Blink in **"Start"**.



Ogni schizzo contiene sempre i metodi **"setup"** e **"loop"**. Il primo viene inizialmente eseguito e in genere utilizzato per inizializzare i pin e l'hardware collegato. Il metodo loop viene invece ripetuto in modo permanente e contiene quasi tutte le altre funzioni.

Il LED interno alla scheda è stato selezionato per qualche tempo tramite la variabile propria dell'IDE **"LED_BUILTIN"**. Poiché i file core ESP32 per l'IDE Arduino sono ancora in fase di sviluppo e il layout dei pin varia a seconda del produttore, questa variabile non funziona qui. Il LED interno alla scheda della scheda di sviluppo ESP32 AZ-Delivery si trova sul pin 1. Modificare lo sketch come nella figura.

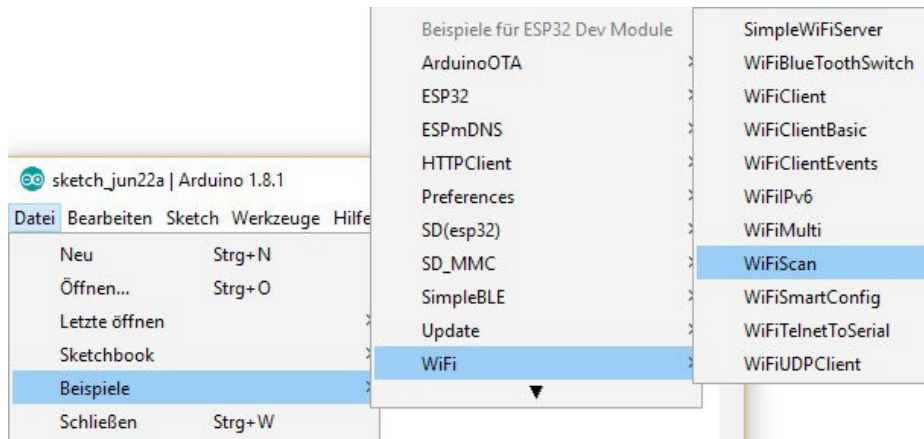


La seconda icona (freccia destra) sotto la barra dei comandi carichi lo schizzo sul NodeMCU.

Se il caricamento ha avuto successo, il LED della tua scheda ora lampeggia ogni secondo.

Ce l'hai fatta! Complimenti!

Successivamente, è necessario esaminare gli script esistenti per ESP32, ad esempio "WifiScan". Uguagliare la velocità di trasmissione del codice con le impostazioni, poi carica lo schizzo sulla scheda di sviluppo. Alcuni secondi dopo dovresti vedere un elenco di tutti i tuoi punti di accesso WLAN con la rispettiva forza del segnale.



Con ESP32 Development Board, insieme al codice Arduino, puoi fare molto di più. Inizia la tua ricerca di ulteriori esempi della libreria Arduino e sul web, per esempio su:

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

Naturalmente, visita il nostro negozio online per il supporto dell'hardware:

<https://az-delivery.de>

Impressum

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