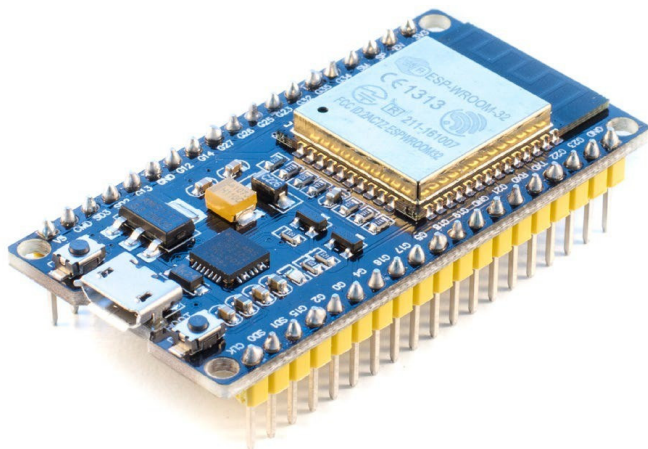


Bienvenido!

Muchas gracias por comprar nuestra **AZ-Delivery ESP-32 Development-Board**! En las siguientes páginas le guiaremos en los primeros pasos del proceso de configuración hasta los primeros scripts.

Le deseamos mucha diversión!



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

El **chip ESP32** es un potente sucesor del muy popular **ESP8266**, que está instalado, por ejemplo, en el **AZ-Delivery NodeMCUs Amica V2** y **Lolin V3**. El cambio más notable de este chip es la conectividad Bluetooth 4.2 BLE agregada. La **AZ-Delivery ESP-32 Development-Board** es el compañero perfecto para una operación flexible en el Internet de las cosas (Internet of Things).

Descripción de las características más importantes

- » Programación a través de un cable micro USB-B
- » Fuente de alimentación a través de:
 - » Micro USB-B en el puerto USB de la computadora
 - » Micro USB-B en el adaptador de corriente USB de 5V
- » Procesador ESP-WROOM-32
 - » WLAN 802.11 b/g/n y Bluetooth 4.2 / BLE
 - » 160MHz Tensilica L108 32 bits CPU de doble núcleo
 - » 512 KB SRAM y 16 MB de memoria flash
- » 32 pines de E / S digitales (3,3V!)
- » 6 pines analógicos a digitales
- » 3x UART, 2x SPI, 2x I²C
- » CP2102 Interfaz USB a UART
- » Programable a través de Arduino Code, Lua, MicroPython,...

En las siguientes páginas encontrará información sobre la

» **Instalación del controlador y preparación del Arduino IDE,**

y una guía para

» **el primer script por Arduino Code**

Todos los Links en un vistazo

Controlador:

- » 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 para ESP: <https://github.com/Nicholas3388/LuaNode>
- » MicroPython Firmware para 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/>

Otras herramientas:

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

Información interesante de AZ-Delivery

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

Instalación del controlador

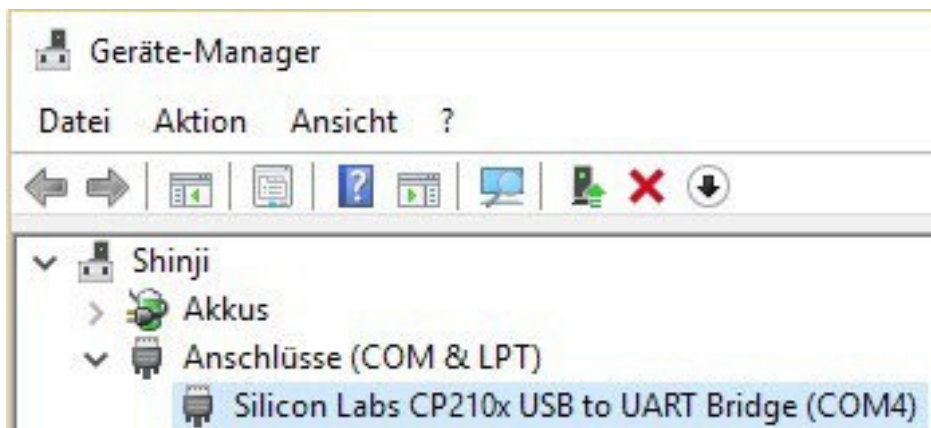
Conecte la **AZ-Delivery ESP-32 Development-Board** a su computadora a través de un cable micro USB. El microcontrolador necesita un **CP2102-Chip** para la interfaz USB, que Windows reconoce automáticamente por los sistemas MacOS.

Si este no es el caso, descargue el controlador actual desde el siguiente enlace y luego descomprímalo:

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

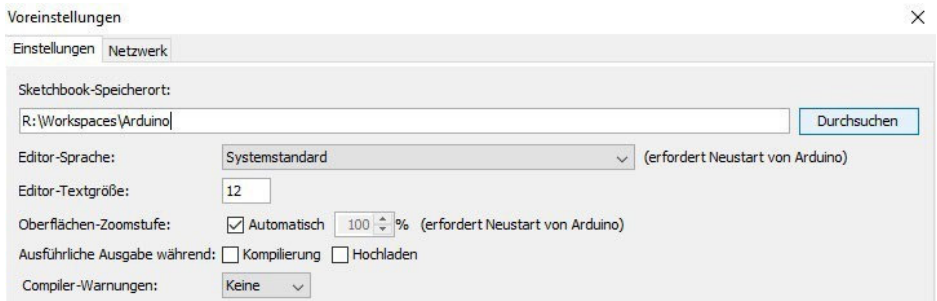
En Windows, simplemente instálelo ejecutando el "**CP210xVCPInstaller_x86.exe**" o "**CP210xVCPInstaller_x64.exe**" dependiendo de su sistema. Como usuario de Mac, debe instalar el archivo DMG en su archivo cargado.

Después de volver a conectar NodeMCU, se debe reconocer como un dispositivo "**Silicon Labs CP210x USB to UART Bridge**" (Windows).



Instalación de la ESP32 Development Board

Visite la página <https://www.arduino.cc/en/Main/Software> y descargue la última versión para su sistema operativo. Alternativamente, puede registrarse en Arduino Web-Editor y seguir las instrucciones fáciles de entender provistas allí. Los siguientes primeros pasos usan la versión de escritorio para Windows.



Una vez que el programa ha comenzado, se debe establecer la ubicación de almacenamiento del primer Sketchbook en **Archivo > Preferencias**, por ejemplo, en **Mis documentos\Arduino**. De esta manera, los scripts de Arduino, llamados **"Sketches"** se almacenarán en el lugar que usted prefiera.

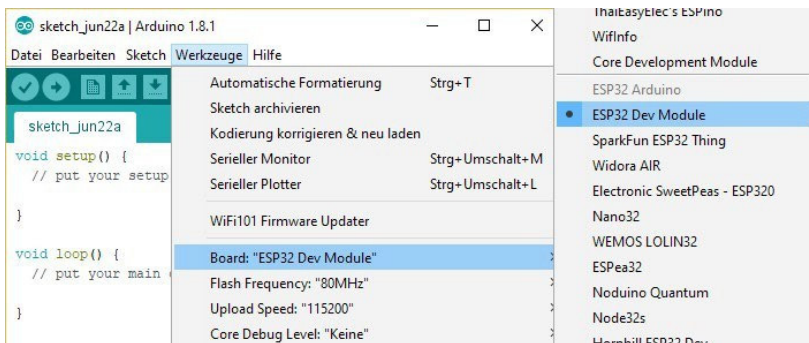
El **ESP32** no es parte del repertorio estándar del IDE, y no se encuentra en el Board-Manager. Descargue los archivos Arduino Core para el controlador:

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

Una vez hecho esto, extraiga el contenido del directorio **"arduino-esp32-master"** en la carpeta del hardware de su Arduino-Sketchbook (por ejemplo, Mis documentos \ ...) \ **Arduino \ hardware \ espressif \ esp32 **. Luego, inicie la aplicación **"get.exe"** en el directorio de herramientas y espere la descarga completa de los archivos necesarios.

Datei Start Freigegeben Ansicht			
< > v ↑ > 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

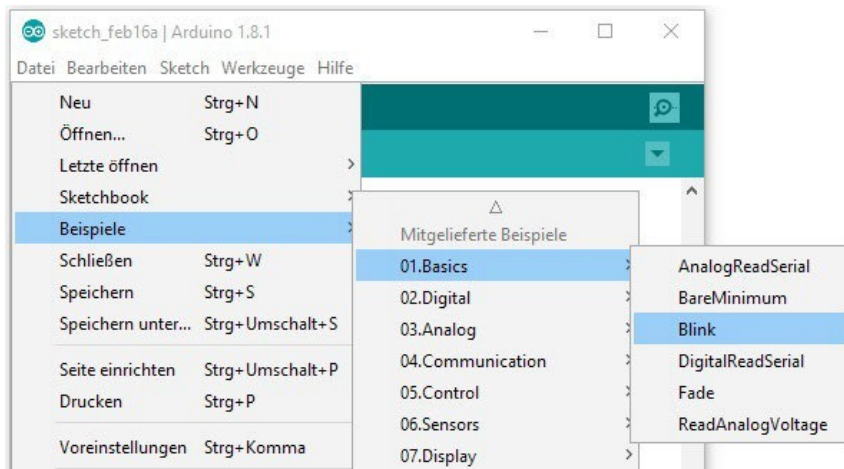
Luego, abra Arduino IDE y en "**Herramientas>Board**" puede seleccionar el "**ESP32 Dev Module**", más la frecuencia flash de **80MHz** y una velocidad en baudios de al menos "**115200**".



El primer Script

Mientras que en la mayoría de los lenguajes de programación, la primera señal de éxito es la aparición de la frase "Hello World!", cuando se trata de Arduinos y otros microcontroladores, sin embargo, el primer signo de éxito es el parpadeo del LED de la placa interna. En conformidad, el script se llama "**Blink**".

»Inicie el Arduino IDE y abra el Blink-Script en "**Start**".



Cada Sketch contiene siempre los métodos "**setup**" y "**loop**". El primero se ejecuta al inicio y se usa normalmente para inicializar pines y el hardware conectado. El método loop se repite de forma permanente y, por lo tanto, contiene casi todas las demás funciones.

El LED interno de la placa se ha seleccionado automáticamente durante algún tiempo a través de la variable del propio IDE "**LED_BUILTIN**". Dado que los archivos ESP32-Core para el Arduino IDE todavía están en desarrollo y los diseños de los pines varían según el fabricante, esta variable no funciona aquí. El LED interno de la placa de **AZ-Delivery ESP32 Development Board** está en el pin 1. Cambie el sketch como en la imagen del medio.

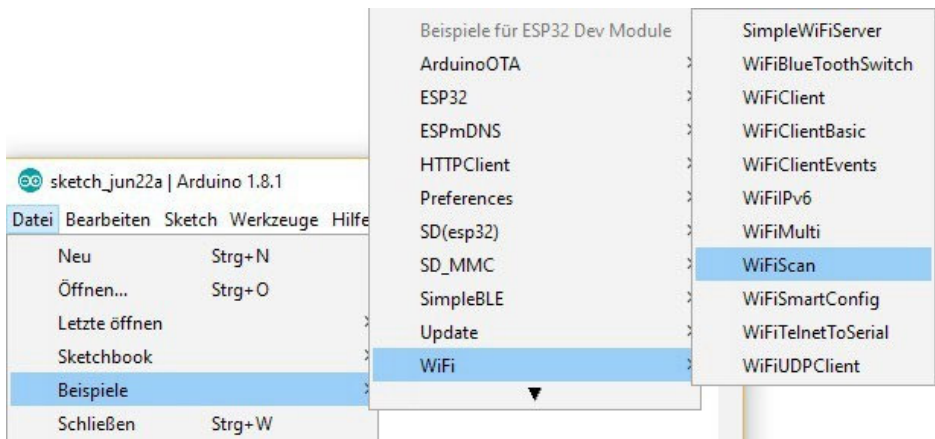


Con el segundo ícono (la flecha que se muestra a la derecha) debajo de la barra de comandos, puede cargar el sketch en el NodeMCU.

Si la carga fue exitosa, el LED de su placa comenzará a parpadear cada segundo.

¡Lo hizo! ¡Felicidades!

A continuación, debe ver los scripts existentes para el ESP-32, por ejemplo "WifiScan". Compare las velocidades en baudios del código con su configuración. Suba el sketch a la Development Board. Finalmente, después de unos segundos, debería visualizar una lista de todos los puntos de acceso WLAN disponibles que se encuentran en su entorno, así como la respectiva intensidad de señal.



Gracias al Arduino Code, puede hacer mucho más con la placa ESP32 Development Board. Comience la búsqueda de más posibilidades, tomando como referencia los otros ejemplos de sketches de la biblioteca Arduino y en la web, por ejemplo en: <http://michaelsarduino.blogspot.de/search?q=8266>.

Para soporte de hardware, nuestra tienda en línea siempre está a su disposición:

<https://az-delivery.de>

Impressum

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