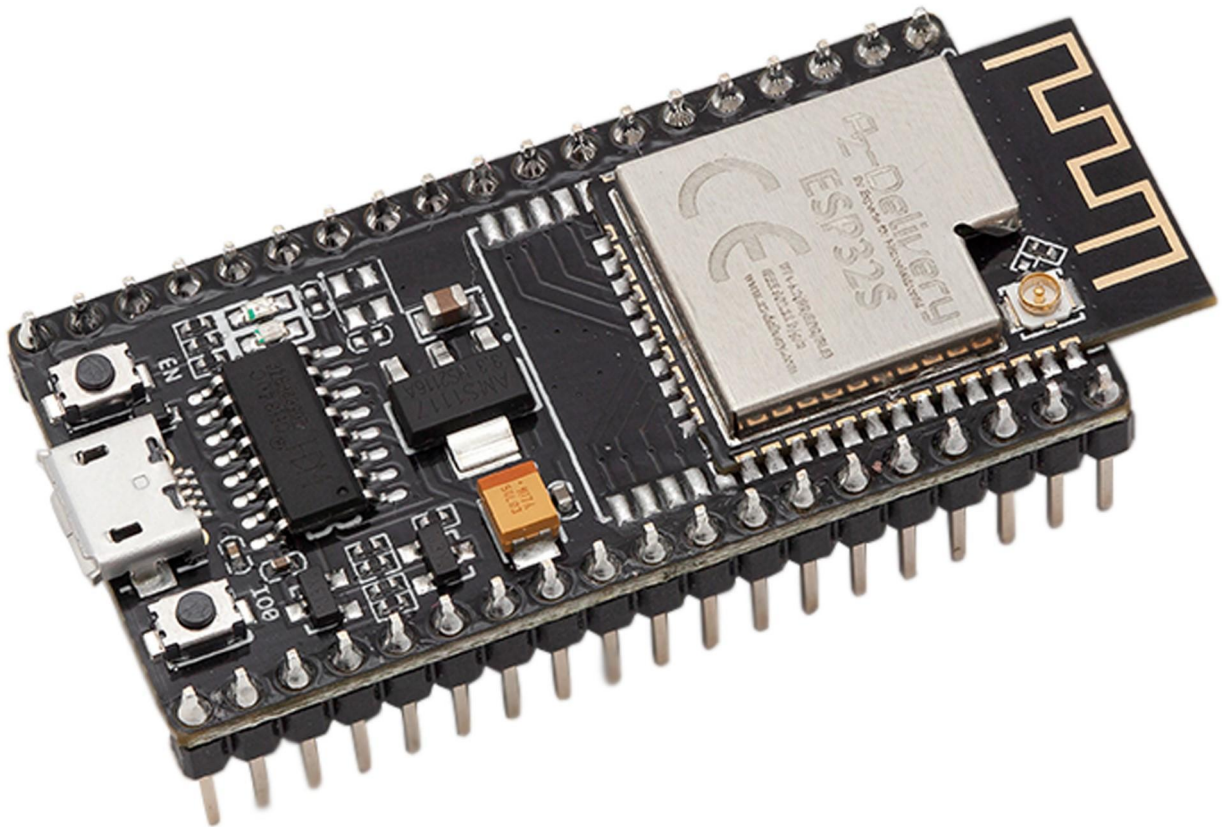




Herzlich willkommen.

Vielen Dank, dass Sie sich für unser AZ-Delivery NodeMCU ESP32S Kit entschieden haben. Auf den folgenden Seiten stellen wir Ihnen vor, wie Sie dieses praktische Gerät verwenden und konfigurieren.

Viel Spaß





Inhaltsübersicht

Einführung	3
Spezifikationen	4
ESP32S Bausatz	5
Pinout	6
Beschreibung der Eingangs-/Ausgangsstifte	7
Stifte des kapazitiven Berührungssensors	8
Analog-Digital-Wandler-Stifte	9
Digital-Analog-Wandler Pins	9
Echtzeituhr GPIO-Pins	10
PWM-Pins (Pulsbreitenmodulation)	11
I2C-Pins	11
SPI-Pins	12
Strapping Pins	12
HIGH-Stifte im Boot	13
EN	13
USB-serielle Kommunikation	14
WiFi-Kommunikation	15
Bluetooth-Kommunikation	16
Anschluss der Antenne	17
Andere Merkmale	18
Einrichten der Arduino IDE	19
sh arduino-linux-setup.sh user_name	20
Zusätzliche Konfiguration	23
ESP32S NodeMCU Kit Verdrahtungsbeispiel	27
Beispiele für Skizzen	28



Einführung

Das ESP32S NodeMCU Kit ist ein Entwicklungsboard, das um den ESP32-Chip herum aufgebaut ist. Es enthält einen Spannungsregler und eine USB-Programmierschaltung für den ESP32-Chip sowie viele weitere Funktionen.

Für die Anwendungsentwicklung können Sie zwischen Arduino IDE oder ESP-IDF (native Plattform) wählen. Die meisten Benutzer entscheiden sich für die Arduino IDE wegen ihrer Einfachheit und Kompatibilität. Die Nutzergemeinschaft ist sehr aktiv und unterstützt Plattformen wie ESP32.

NodeMCU ESP32S Kit wird mit vorinstallierter Firmware geliefert, die das Arbeiten mit der interpretierten Sprache und das Senden von Befehlen über den seriellen Port (CH340 Chip) ermöglicht. Das ESP32-Board ist eine der am häufigsten verwendeten Plattformen für Internet of Things (IoT)-Projekte.

Das ESP32S Kit NodeMCU Board ist speziell für die Arbeit auf dem Breadboard entwickelt worden. Es ist mit einem Spannungsregler ausgestattet, der die Stromversorgung direkt über den USB-Anschluss ermöglicht. Die Eingangs-/Ausgangsstifte arbeiten mit 3,3 V. Der CH340-Chip ist für die USB-Seriell-Kommunikation zuständig.



Spezifikationen

Versorgungsspannung (USB)	5V DC
Eingangs-/Ausgangsspannung	3.3V DC
Erforderlicher Arbeitsstrom	min. 500mA
SoC	ESP32S
CPU	Xtensa®32-bit LX6 MCUs
Taktfrequenzbereich	80MHz / 160MHz / 240MHz
ROM	448kB
RAM	520kB
Externer Flash-Speicher	4MB
I/O pins	34
ADC Kanäle	18
ADC Auflösung	12-bit
DAC Kanäle	2
DAC Auflösung	8-bit
Kommunikationsschnittstellen	SPI, I2C, I2S, CAN, UART
Wi-Fi Protokolle	802.11 b/g/n (802.11n up to 150 Mbps)
Wi-Fi Frequenz	2.4 GHz - 2.5 GHz
Bluetooth	V4.2 - BLE and Classic Bluetooth
Drahtlose Antenne	PCB, Anschluss für externe Antenne
Abmessungen	55x26x13mm(2.1x1x0.5in)



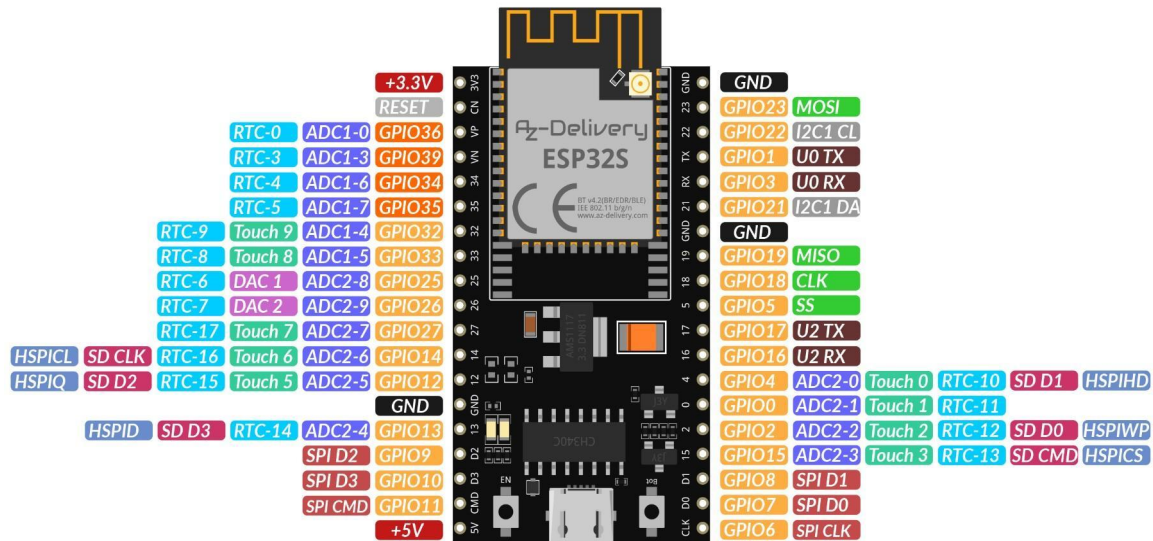
ESP32S Bausatz

Die NodeMCU ESP32S Kit-Serie ist ein erschwingliches Wi-Fi-Modul, das sich für DIY-Projekte im Bereich Internet of Things (IoT) eignet. Dieses Modul verfügt über viele GPIOs und unterstützt eine Vielzahl von Protokollen wie SPI, I2C, I2S, UART und mehr. Das Beste daran ist, dass er mit einem drahtlosen Netzwerk ausgestattet ist, was ihn von anderen bekannten Mikrocontrollern unterscheidet. Das bedeutet, dass Sie Geräte einfach und kostengünstig über Wi-Fi und Bluetooth® aus der Ferne steuern und überwachen können. Der ESP32S ist ein System-on-Chip (SoC), der eine 32-Bit-Dual-Core-MCU, digitale Standard-Peripherieschnittstellen, Antennenschalter, RF-Balun, Leistungsverstärker, rauscharmen Empfangsverstärker, Filter und Power-Management-Module in einem kleinen Gehäuse integriert. Er bietet 2,4-GHz-Wi-Fi (802.11 b/g/n, unterstützt Geschwindigkeiten von bis zu 150 MB/s), BLE und klassische drahtlose Bluetooth®-Kommunikation, 34 I/O-Pins, I2C- und I2S-Schnittstellen, ADC (Analog-Digital-Wandlung), DAC (Digital-Analog-Wandlung), SPI-Schnittstelle, UART auf dedizierten Pins und PWM (Pulsweitenmodulation).

Der Prozessorkern mit der Bezeichnung LX6 basiert auf dem 32-Bit LX6 Dual-Core Xtensa® Prozessor-Controller und arbeitet mit einer Frequenz zwischen 80 und 240MHz. Er verfügt über ein 448kB großes Boot-ROM, 520kB On-Chip-SRAM und bis zu 16MB externen Flash-Speicher, auf den über die SPI-Schnittstelle zugegriffen werden kann. Folglich haben die Anbieter viele kompakte PCB-Module auf der Basis des ESP32S-Chips entwickelt. Einige dieser Module haben spezifische Bezeichnungen.

Pinout

Das ESP32S NodeMCU Kit hat 38 Pins. Die Pinbelegung ist in der folgenden Abbildung dargestellt



Eine detaillierte Beschreibung der Pins und E/A-Funktionen finden Sie im Datenblatt, das unter [link](#).

NOTE: Der absolute Maximalstrom, der von einem GPIO gezogen wird, beträgt 40mA gemäß dem Abschnitt "Empfohlene Betriebsbedingungen" im ESP32-Datenblatt.



Beschreibung der Eingangs-/Ausgangsstifte

Wie ein normales ATmega-Board hat das ESP32S NodeMCU-Kit digitale Eingangs-/Ausgangs-Pins (GPIO - General Purpose Input/Output Pins). Diese digitalen Ein- und Ausgänge arbeiten mit 3,3 V.

WARNUNG: Die 5V Spannung darf nicht an einen Pin des ESP32 Chips angeschlossen werden!

Die Pins sind nicht 5V-tolerant, das Anlegen von mehr als 3,3V an einem Pin führt zur Zerstörung des Chips.

Die GPIO-Pins 34 bis 39 sind GPI-Eingangsstifte, die keine internen Pull-up- oder Pull-down-Widerstände haben. Sie können nicht als Ausgänge verwendet werden. Verwenden Sie daher nur diese Pins als Eingänge: GPIO 34, GPIO 35, GPIO 36, GPIO 39.

Der ESP32S-Chip verfügt über ein integriertes SPI-Flash. Die Pins GPIO6 bis GPIO 11 sind auf bestimmten ESP32-Entwicklungsboards verfügbar. Diese Pins sind mit dem On-Chip-SPI-Flash verbunden und werden für andere Zwecke nicht empfohlen.

GPIO 6 (SCK/CLK), GPIO 7 (SDO/SD0), GPIO 8 (SDI/SD1), GPIO 9 (SHD/SD2), GPIO 10 (SWP/SD3), GPIO 11 (CSC/CMD).

Im Wesentlichen stehen folgende GPIOs für Ihr Projekt zur Verfügung, wenn sie nicht für spezifische Schnittstellen verwendet werden, auch mit Pull-up- oder Pull-down-Widerständen: GPIOs 0, 2, 4, 5, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 25, 26, 27, 32, 33.



Stifte des kapazitiven Berührungssensors

Der ESP32 hat 10 interne kapazitive Touchsensoren. Die kapazitiven Touch-Pins können auch verwendet werden, um den ESP32 aus dem Tiefschlaf zu wecken. Diese internen Berührungssensoren sind an folgende GPIOs angeschlossen: T0 (GPIO 4), T1 (GPIO 0), T2 (GPIO 2), T3 (GPIO 15), T4 (GPIO 13), T5 (GPIO 12), T6 (GPIO 14), T7 (GPIO 27), T8 (GPIO 33), T9 (GPIO 32).



Analog-Digital-Wandler-Stifte

Der ESP32 hat 18x12 Bit ADC (Analog-Digital-Wandler) Eingangskanäle (während der ESP8266 nur 1x10 Bit ADC hat). Dies sind die GPIOs, die als ADCs verwendet werden können, und die entsprechenden Kanäle:

ADC1_CH0 (GPIO 36), ADC1_CH1 (GPIO 37), ADC1_CH2 (GPIO 38),
ADC1_CH3 (GPIO 39), ADC1_CH4 (GPIO 32), ADC1_CH5 (GPIO 33),
ADC1_CH6 (GPIO 34), ADC1_CH7 (GPIO 35), ADC2_CH0 (GPIO 4),
ADC2_CH1 (GPIO 0), ADC2_CH2 (GPIO 2), ADC2_CH3 (GPIO 15),
ADC2_CH4 (GPIO 13), ADC2_CH5 (GPIO 12), ADC2_CH6 (GPIO 14),
ADC2_CH7 (GPIO 27), ADC2_CH8 (GPIO 25), ADC2_CH9 (GPIO 26).

Digital-Analog-Wandler Pins

Der ESP32 verfügt über 2x8-Bit-DAC-Kanäle (Digital-Analog-Wandler), die digitale Signale in analoge Spannungssignale umwandeln.

Dies sind die DAC-Kanäle: DAC1 (GPIO25), DAC2 (GPIO26).



Echtzeituhr GPIO-Pins

Der ESP32 verfügt über eine RTC (Echtzeituhr) GPIO Unterstützung. Die GPIOs, die an das RTC-Subsystem mit niedrigem Stromverbrauch geleitet werden, können verwendet werden, wenn sich der ESP32 im Tiefschlaf befindet. Diese RTC GPIOs können verwendet werden, um den ESP32 aus dem Tiefschlaf zu wecken, wenn der Ultra Low Power (ULP) Coprozessor läuft. Die folgenden GPIOs können als externe Wake-up-Quelle verwendet werden: RTC_GPIO0 (GPIO36), RTC_GPIO3 (GPIO39), RTC_GPIO4 (GPIO34), RTC_GPIO5 (GPIO35), RTC_GPIO6 (GPIO25), RTC_GPIO7 (GPIO26), RTC_GPIO8 (GPIO33), RTC_GPIO9 (GPIO32), RTC_GPIO10 (GPIO4), RTC_GPIO11 (GPIO0), RTC_GPIO12 (GPIO2), RTC_GPIO13 (GPIO15), RTC_GPIO14 (GPIO13), RTC_GPIO15 (GPIO12), RTC_GPIO16 (GPIO14), RTC_GPIO17 (GPIO27).



PWM-Pins (Pulsbreitenmodulation)

Der PWM (Pulsweitenmodulation) Controller des ESP32 hat 16 unabhängige Kanäle, die so konfiguriert werden können, dass sie PWM-Signale mit unterschiedlichen Eigenschaften erzeugen. Alle Pins, die als Ausgänge fungieren können, können als PWM-Pins verwendet werden (GPIOs 34 bis 39 können keine PWM erzeugen). Um ein PWM-Signal zu konfigurieren, müssen diese Parameter im Code definiert werden: Frequenz des Signals, Tastverhältnis, PWM-Kanal, GPIO, an dem das Signal ausgegeben werden soll.

I2C-Pins

Der ESP32 hat zwei I2C-Kanäle und jeder GPIO-Pin kann als SDA oder SCL eingestellt werden. Wenn Sie den ESP32 mit der Arduino IDE verwenden, sind die Standard I2C Pins:

GPIO 21 (SDA), GPIO 22 (SCL).

SPI-Pins

Standardmäßig ist die Pinbelegung für die SPI-Pins wie folgt:

SPI	MOSI	MISO	CLK	CS
VSPI	GPIO 23	GPIO 19	GPIO 18	GPIO 5
HSPI	GPIO 13	GPIO 12	GPIO 14	GPIO 15

Strapping Pins

Die folgenden Pins werden verwendet, um den ESP32 in den Bootloader- oder Flash-Modus zu versetzen: GPIO 0, GPIO 2, GPIO 4, GPIO 5 (muss während des Bootvorgangs HIGH sein), GPIO 12 (muss während des Bootvorgangs LOW sein), GPIO 15 (muss während des Bootvorgangs HIGH sein).

Die meisten Entwicklungsboards setzen die Pins in den richtigen Zustand für den Flash- oder Boot-Modus. Wenn einige Peripheriegeräte mit den Boot-Pins verbunden sind und die IDE keinen Code laden oder den ESP32 flashen kann, kann es daran liegen, dass diese Peripheriegeräte den ESP32 daran hindern, in den richtigen Modus zu gelangen. Nach einem Neustart, Flashen oder Booten funktionieren diese Pins wie erwartet. Es gibt einen Dokumentationsleitfaden zur Auswahl des Bootmodus unter [link](#). Weitere und ausführlichere Erklärungen sind nicht im Rahmen dieses E-Books möglich, daher verweisen wir im Folgenden auf das Esptool-Datenblatt oder die Dokumentation [link](#).



HIGH-Stifte im Boot

Einige GPIOs ändern beim Booten oder Zurücksetzen ihren Zustand auf HIGH oder geben PWM-Signale ab. Das bedeutet, dass wenn die Ausgänge mit diesen GPIOs verbunden sind, dies zu unerwarteten Ergebnissen führen kann, wenn der ESP32 neu startet oder hochfährt.

GPIO 1, GPIO 3, GPIO 5, GPIO 6 bis GPIO 11 (verbunden mit dem eingebauten SPI-Flash-Speicher des ESP32 - nicht empfohlen), GPIO 14, GPIO 15.

EN

Enable (EN) ist der Freigabe-Pin des 3,3-V-Reglers. Er hat einen Pull-up-Status und muss mit Masse verbunden werden, um den 3,3-V-Regler zu deaktivieren. Das bedeutet, dass dieser Pin mit einem Taster verbunden werden kann, um z.B. Ihren ESP32 zurückzusetzen.



USB-serielle Kommunikation

Das ESP32S NodeMCU-Kit hat einen microUSB-Anschluss. Es basiert auf dem CH340-Chip, der die serielle Kommunikation zwischen USB und UART ermöglicht. Der Chip hat die Funktion eines virtuellen COM-Ports (VCP), der in PC-Anwendungen als COM-Port erscheint. Um den ESP32 nutzen zu können, muss der entsprechende Treiber installiert sein.



WiFi-Kommunikation

Das NodeMCU ESP32S Kit hat eine integrierte Wi-Fi Kommunikationsschnittstelle und kann in drei verschiedenen Modi betrieben werden: Wi-Fi-Station, Wi-Fi-Zugangspunkt und beides gleichzeitig. Es unterstützt die folgenden Funktionen:

- 802.11b- und 802.11g-Datenübertragungsraten
- 802.11n MCS0-7 mit 20MHz und 40MHz Bandbreite
- 802.11n MCS32
- 802.11n 0,4µS Schutzintervall
- Datenrate bis zu 150 Mbps
- Empfangender STBC 2x1
- Bis zu 20 dBm Sendeleistung
- Einstellbare Sendeleistung
- Antennenvielfalt und -auswahl (Hardware durch Software gesteuert)



Bluetooth-Kommunikation

Das ESP32S NodeMCU-Kit hat ein integriertes Bluetooth-Funkgerät und unterstützt die folgenden Funktionen:

- Sendeausgangsleistungen der Klassen 1, 2 und 3 und ein dynamischer Regelbereich von über 30 dB.
- $\pi/4$ DQPSK- und 8 DPSK-Modulation
- Hohe Empfindlichkeit des NZIF-Empfängers mit einem Dynamikbereich von über 98 dB
- Betrieb der Klasse 1 ohne externe Beschallungsanlage
- Interner SRAM ermöglicht Datenübertragung mit voller Geschwindigkeit, Sprach-/Datenmischung und vollen Piconet-Betrieb
- Logik für Vorwärtsfehlerkorrektur, Header-Fehlerprüfung, Zugriffscode-Korrelation, CRC, Demodulation, Erzeugung von Verschlüsselungs-Bitströmen, Whitening und Formung von Übertragungsimpulsen
- ACL, SCO, eSCO und AFH
- A-law-, μ -law- und CVSD-Digital-Audio-CODEC in PCM-Schnittstelle
- SBC-Audio-CODEC
- Energiemanagement für Anwendungen mit geringem Stromverbrauch
- SMP mit 128-Bit-AES



Darüber hinaus unterstützt das Bluetooth-Funkgerät die folgenden Kommunikationsschnittstellenprotokolle:

- UART HCI Schnittstelle, bis zu 4Mbps
- SDIO / SPI HCI-Schnittstelle
- I2C-Schnittstelle
- PCM / I2S-Audio-Schnittstelle.

Anschluss der Antenne

Zusätzlich zur PCB-Antenne verfügt das ESP32S NodeMCU-Kit über eine P.FL-Antennenbuchse.

Diese Buchse wird auch als IPEX, IPAX, IPX, AMC, MHF oder UMCC bezeichnet. Um diese Buchse zu aktivieren, muss der kleine Widerstand (00hm/SMD1005) neben der Buchse aufgelöst werden.

Die Antenne muss eine 2,4-GHz-Leiterplattenantenne mit einem maximalen Gewinn von 2 dBm sein.



Andere Merkmale

Der ESP32S-Chip hat einen integrierten Hall-Effekt-Sensor, der Änderungen des Magnetfeldes in seiner Umgebung erkennt.

Der Hall-Sensor basiert auf einem N-Träger-Widerstand. Wenn sich der Chip im Magnetfeld befindet, entwickelt der Hall-Sensor eine kleine Spannung über dem Widerstand, die vom Analog-Digital-Wandler (ADC) direkt gemessen oder vom analogen Vorverstärker mit sehr geringem Rauschen verstärkt und dann vom ADC gemessen werden kann.

Der Temperatursensor erzeugt eine Spannung, die mit der Temperatur variiert. Die Spannung wird intern durch einen Analog-Digital-Wandler in einen digitalen Code umgewandelt. Der Temperatursensor hat einen Bereich von -40°C bis 125°C . Da die Verschiebung des Temperatursensors von Chip zu Chip aufgrund von Prozessvariationen variiert, zusammen mit der durch den Wi-Fi-Schaltkreis selbst erzeugten Wärme (die die Messungen beeinflusst), eignet sich der interne Temperatursensor nur für Anwendungen, die eher Temperaturänderungen als absolute Temperaturen erfassen, sowie für Kalibrierungszwecke. Wenn der Benutzer jedoch den Temperatursensor kalibriert und das Gerät in einer Anwendung mit minimaler Stromversorgung verwendet, könnten die Ergebnisse ausreichend genau sein.

Einrichten der Arduino IDE

Wenn die Arduino IDE nicht installiert ist, folgen Sie den Anweisungen [link](#) und laden Sie die Installationsdatei für das von Ihnen gewählte Betriebssystem herunter. Die für dieses eBook verwendete Version der Arduino IDE ist **1.8.19**.



Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

Windows app Win 8.1 or 10 [Get](#) 

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)

Windows-Benutzer doppelklicken auf die heruntergeladene .exe-Datei und folgen den Anweisungen im Installationsfenster.

Az-Delivery

Für *Linux-Benutzer* laden Sie eine Datei mit der Erweiterung *.tar.xz* herunter, die entpackt werden muss. Wechseln Sie nach dem Entpacken in das extrahierte Verzeichnis und öffnen Sie das Terminal in diesem Verzeichnis. Führen Sie zwei *.sh*-Skripte aus, das erste heißt *arduino-linux-setup.sh* und das zweite heißt *install.sh*.

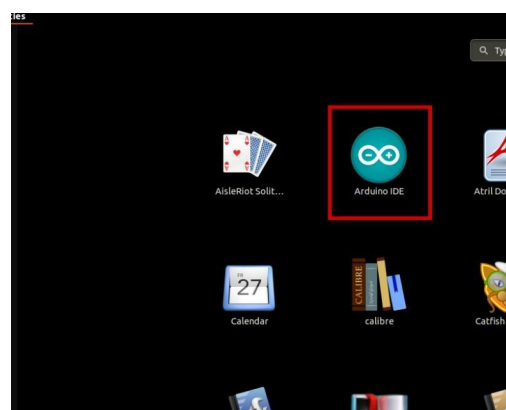
Um das erste Skript im Terminal auszuführen, öffnen Sie das Terminal im extrahierten Verzeichnis und führen Sie den folgenden Befehl aus:

```
sh arduino-linux-setup.sh user_name
```

user_name - ist der Name eines Superusers im Linux-Betriebssystem. Sie müssen ein Passwort für den Superuser eingeben, wenn Sie den Befehl starten. Warten Sie ein paar Minuten, bis das Skript alles abgeschlossen hat.

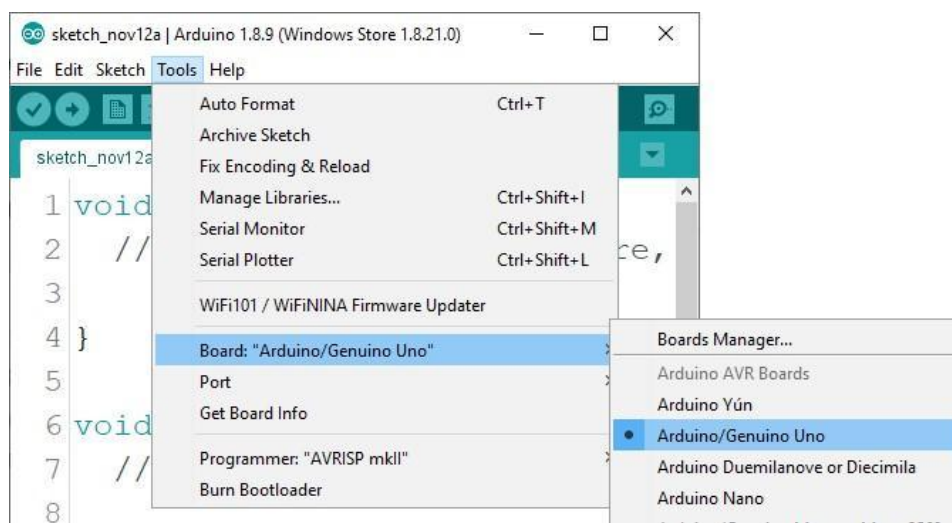
Das zweite Skript, *install.sh*, muss nach der Installation des ersten Skripts verwendet werden. Führen Sie den folgenden Befehl im Terminal aus (entpacktes Verzeichnis): *sh install.sh*

Nachdem Sie diese Skripte installiert haben, gehen Sie in den Bereich Alle Apps, in dem die Arduino IDE installiert ist.



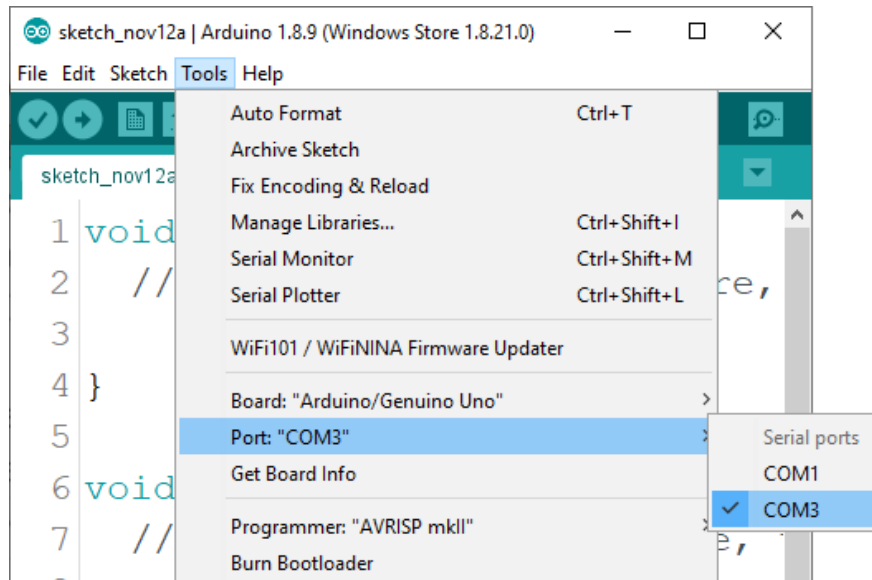
Fast jedes Betriebssystem wird mit einem vorinstallierten Texteditor geliefert (z. B. Windows mit *Notepad*, *Linux Ubuntu* mit *Gedit*, *Linux Raspbian* mit *Leafpad* usw.). Alle diese Texteditoren sind für den Zweck des E-Books vollkommen ausreichend.

Als Nächstes müssen Sie prüfen, ob Ihr PC eine Mikrocontroller-Karte erkennen kann. Öffnen Sie die neu installierte Arduino IDE und gehen Sie zu Tools > Board > {Ihr Boardname hier} {Ihr Boardname hier} {Ihr Boardname hier} sollte der Arduino/Genuino Uno sein, wie Sie im Bild unten sehen können:



Sie müssen den Port auswählen, an den die Mikrocontrollerplatine angeschlossen ist. Gehen Sie zu: Extras > Anschluss > {Anschlussname hierher}. Wenn die Mikrocontrollerplatine an den USB-Anschluss angeschlossen ist, wird der Anschlussname im Dropdown-Menü in der Abbildung oben angezeigt.

Wenn Sie die Arduino-IDE unter Windows verwenden, lauten die Anschlussnamen wie folgt:



Für **Linux-Benutzer** lautet der Anschlussname beispielsweise **/dev/ttyUSBx**, wobei x für eine ganze Zahl zwischen 0 und 9 steht.



Zusätzliche Konfiguration

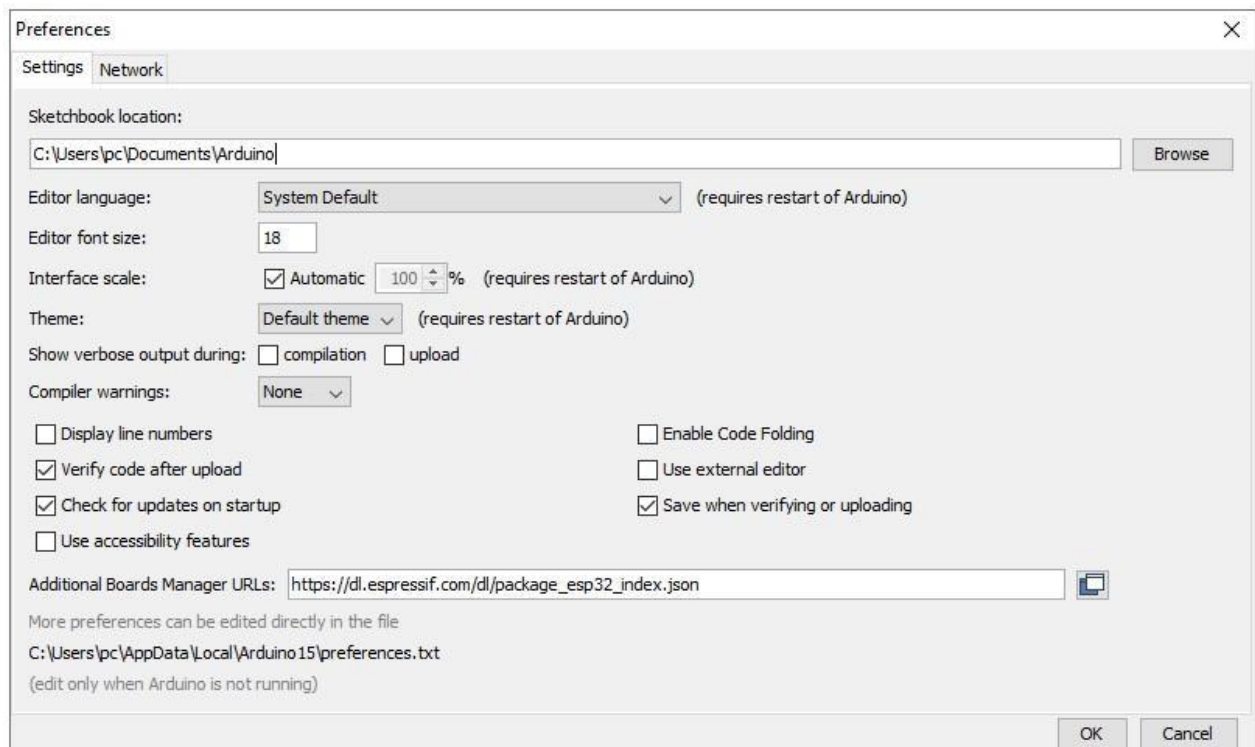
Um das ESP32S NodeMCU-Kit mit der Arduino-IDE zu verwenden, folgen Sie ein paar einfachen Schritten. Vor der Konfiguration der Arduino-IDE muss der Treiber für die USB-serielle Kommunikation installiert werden. Wenn der Treiber nicht automatisch installiert wird, gibt es eine Support-Seite mit Treibern für Windows/Mac oder Linux, aus denen Sie wählen können. Die Treiber können unter der folgenden Adresse heruntergeladen werden [link](#).

Az-Delivery

Um die Unterstützung für die ESP32-Plattform zu installieren, öffnen Sie die Arduino-IDE und gehen Sie zu: Datei > Einstellungen, und suchen Sie das Feld Zusätzliche URLs.

Kopieren Sie dann die folgende URL:

https://dl.espressif.com/dl/package_esp32_index.json



Az-Delivery

Fügen Sie diesen Link in das Feld Zusätzliche URLs ein. Wenn es einen oder mehrere Links in diesem Feld gibt, fügen Sie einfach ein Komma nach dem letzten Link ein, fügen Sie den neuen Link nach dem Komma ein und klicken auf die Schaltfläche OK.



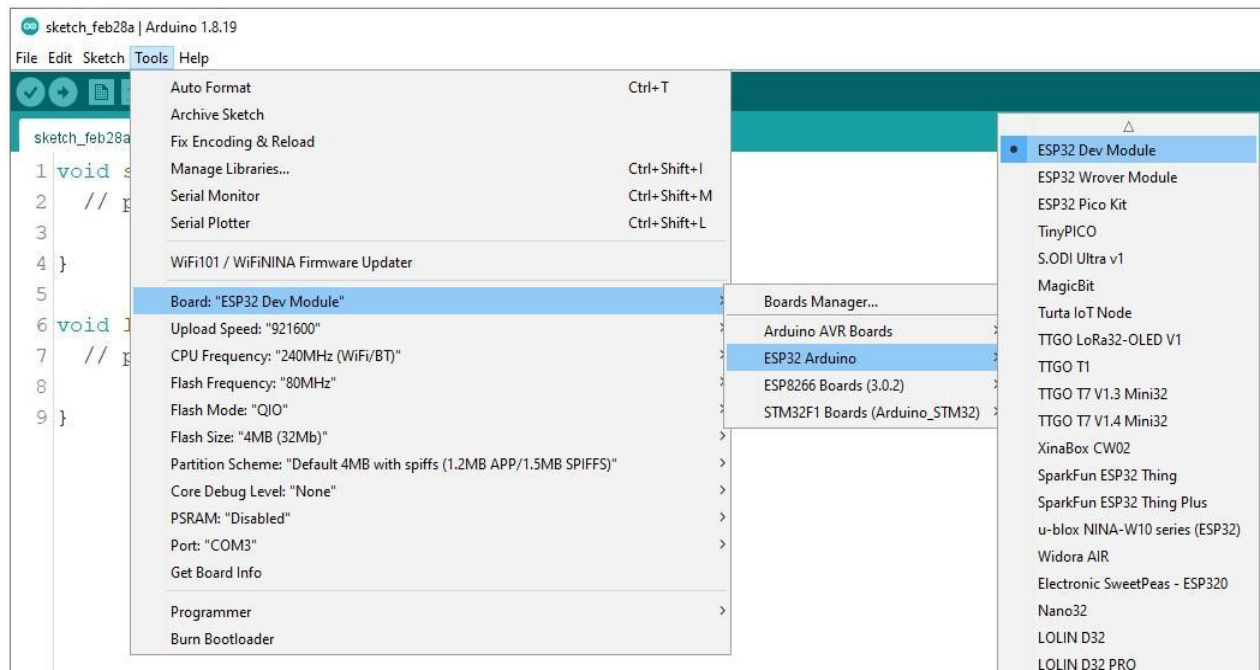
Schließen und öffnen Sie die Arduino-IDE erneut und gehen Sie zu
Werkzeuge > Boards > Board Manager

Wenn sich das neue Fenster öffnet, geben Sie esp32 in das Suchfeld ein und installieren Sie die von Espressif Systems hergestellte Karte esp32, wie in der folgenden Abbildung gezeigt:

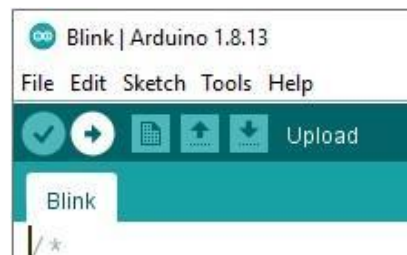


Um das ESP32-Board auszuwählen, gehen Sie zu:

Tools > Board > ESP32 Arduino > ESP32 Dev Module

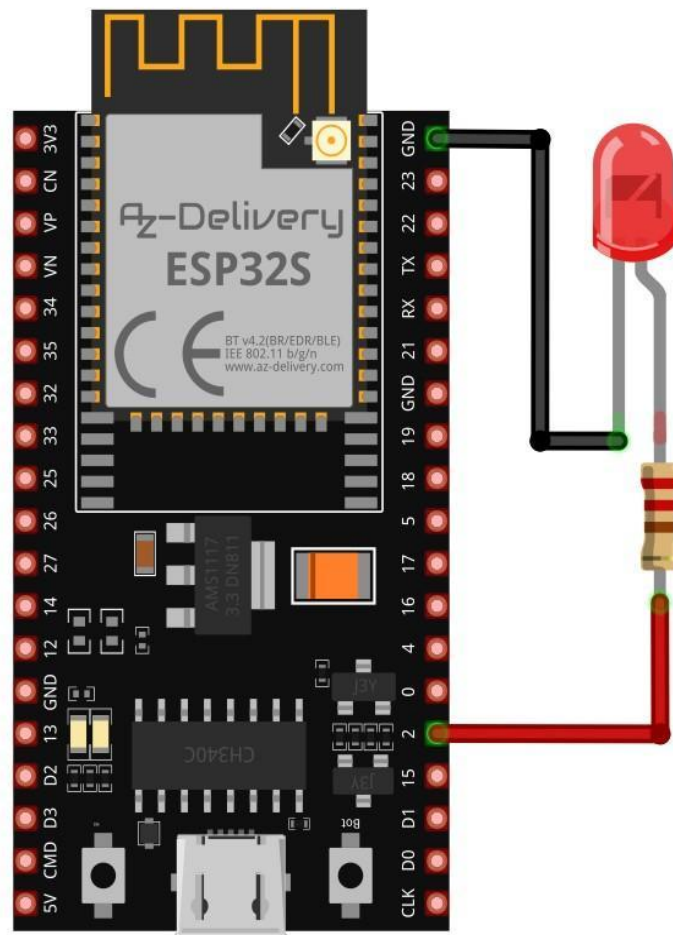


Um den Sketch-Code auf das ESP32-Board hochzuladen, wählen Sie zunächst den Port aus, an dem Sie das Board angeschlossen haben. Gehen Sie zu: **Werkzeuge > Anschluss > {Anschlussname}**



ESP32S NodeMCU Kit Verdrahtungsbeispiel

Verbinden Sie das ESP32S NodeMCU Kit mit einer LED und einem Widerstand wie im folgenden Anschlussdiagramm gezeigt:



NodeMCU ESP32S Kit pin	LED pin	Color del cable
GPIO2 (pin2)	Anode (+) through resistor	Cable rojo
GND	Cathode (-)	Cable negro

Blinking LED

```
int ledPin = 2;

void setup() {
  pinMode(ledPin, OUTPUT); }

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000); }
```

HINWEIS: Obwohl die Variable **LED_BUILTIN** nicht in der ESP32-Firmware enthalten ist, gibt es eine eingebaute blaue LED, die an GPIO2 angeschlossen ist.



PWM - Pulse Width Modulation

```
#define LEDC_CHANNEL_0 0
#define LEDC_TIMER_13_BIT 13
#define LEDC_BASE_FREQ 5000
#define LED_PIN 2

int brightness = 0;
int fadeAmount = 5;

void ledcAnalogWrite(uint8_t channel, uint32_t value, uint32_t valueMax = 255)
{   uint32_t duty = (8191 / valueMax) * min(value,
valueMax);   ledcWrite(channel, duty); }

void setup() {   ledcSetup(LEDC_CHANNEL_0, LEDC_BASE_FREQ,
LEDC_TIMER_13_BIT);   ledcAttachPin(LED_PIN, LEDC_CHANNEL_0); }

void loop() {   ledcAnalogWrite(LEDC_CHANNEL_0,
brightness);   brightness = brightness +
fadeAmount;   if (brightness <= 0 || brightness
>= 255) {       fadeAmount = -fadeAmount;   }
delay(30);
}
```




Jetzt ist es an der Zeit, zu lernen und eigene Projekte zu entwickeln. Sie können dies mit Hilfe vieler Beispielskripte und anderer Anleitungen tun, die Sie im Internet finden können.

Wenn Sie auf der Suche nach hochwertiger Mikroelektronik und Zubehör sind, sind Sie bei der AZ-Delivery Vertriebs GmbH an der richtigen Adresse. Zahlreiche Anwendungsbeispiele, umfassende Installationsanleitungen, E-Books, Bibliotheken und Unterstützung durch unsere technischen Experten sind verfügbar.

<https://az-delivery.de>

Viel Spaß!

Impressum

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