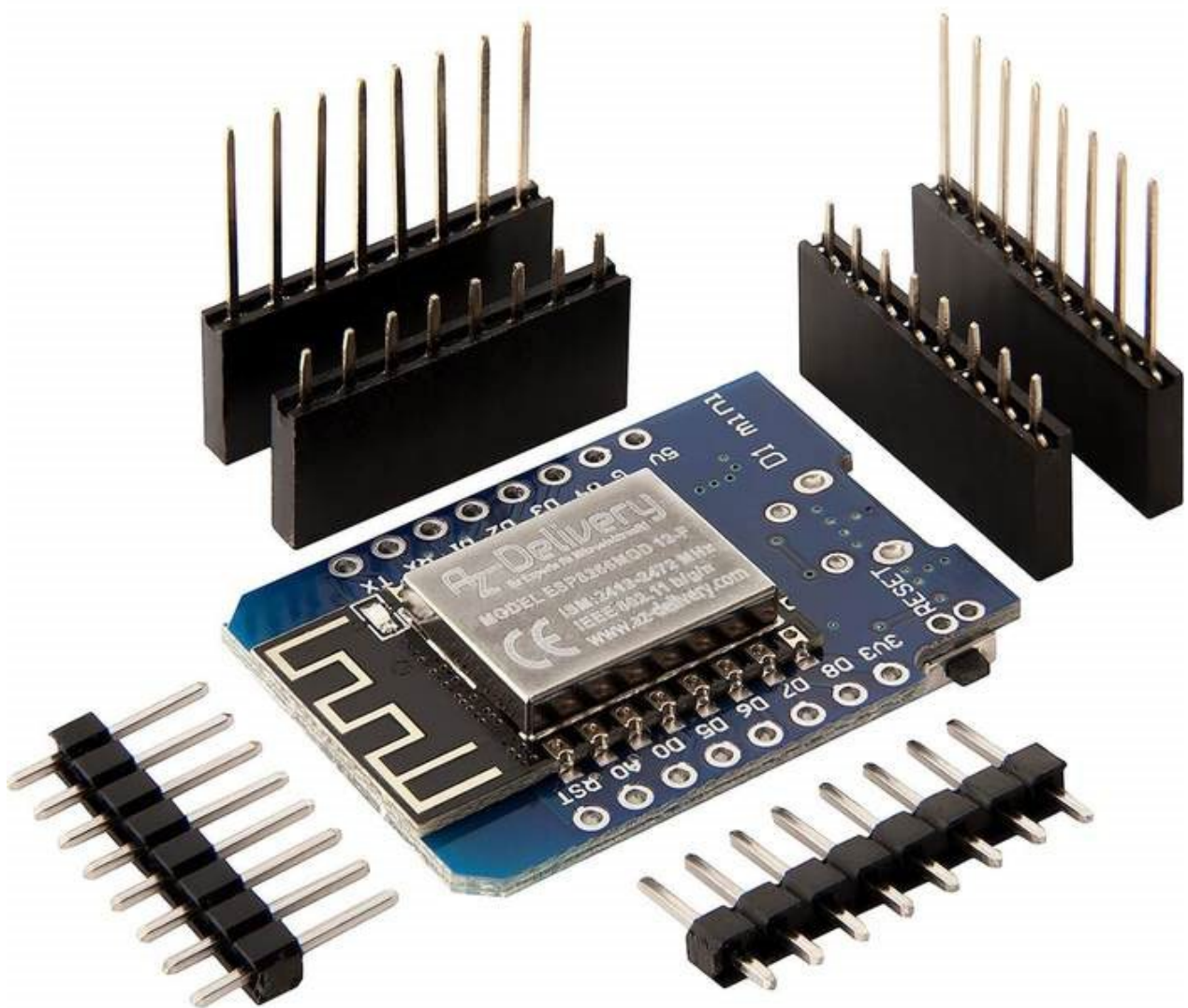


Willkommen!

Vielen Dank, dass Sie sich für unser AZ-Delivery D1 Mini ESP8266-Modul entschieden haben. Auf den folgenden Seiten erfahren Sie, wie Sie dieses handliche Gerät verwenden und einrichten können.

Viel Spaß!



Inhaltsverzeichnis

Introduction.....	3
Specifications of the ESP8266.....	4
The D1 Mini module.....	5
Specifications.....	5
Digital I/O pins.....	6
PWM.....	7
Analog input.....	7
Serial.....	7
The I2C.....	8
SPI.....	8
The pinout.....	9
D1 Mini module - Software.....	10
Digital I/O pins.....	10
Analog input pin	11
Serial communication.....	12
The I2C and SPI interfaces	12
Sharing CPU time with the RF part	13
How to set-up Arduino IDE.....	14
D1 Mini with Arduino IDE	18
Blink + PWM + Serial sketch examples.....	22
Blink sketch example	22
Software PWM sketch example.....	23
Serial communication sketch example	25

Einführung

Das ESP8266-Modul ist ein System on a Chip (SoC). Es besteht aus einem Tensilica L106 32-Bit-Mikrocontroller und einem Wi-Fi-Transceiver. Es hat 11 General Purpose Input/Output Pins, oder kurz GPIO Pins, und einen analogen Eingang. Das bedeutet, dass er wie jeder andere Mikrocontroller programmiert werden kann. Das Beste am ESP8266 ist, dass er über Wi-Fi-Kommunikation verfügt, was bedeutet, dass er mit Wi-Fi oder dem Internet verbunden werden kann, einen Webserver mit echten Webseiten hosten kann, sich mit einem Smartphone verbinden kann, usw. Er unterstützt Netzwerkprotokolle wie Wi-Fi, TCP, UDP, HTTP, DNS, etc.

Das AZ-Delivery D1 Mini-Modul ist eine Entwicklungsplatine auf Basis des ESP8266-Chips. Es hat 11 digitale Eingangs-/Ausgangs-Pins und einen analogen Eingangs-Pin. Alle digitalen E/A-Pins haben Interrupt-, Pwm-, I2C- und 1-Wire-Fähigkeiten, in Software. Der Bereich der analogen Eingangsspannung liegt zwischen 0V und 3,3V DC. Das Modul verwendet einen microUSB-Port und den CH340G-Chip mit einer Programmierschaltung zur Programmierung des ESP8266. Außerdem bietet der microUSB-Port eine Stromversorgung für das Modul.

Es gibt verschiedene Möglichkeiten, das D1-Mini-Modul zu programmieren. Wenn Sie schon einmal mit Arduino-Boards gearbeitet haben, dann ist dies wirklich einfach für Sie. Denken Sie nur daran, dass es nicht auf diese Option beschränkt ist, es gibt viele andere Möglichkeiten, das D1 Mini-Modul zu programmieren (offizielles ESP SDK für C-Programmierung, Lua-Interpreter, MicroPython-Firmware, sind nur wenige von vielen).

Spezifikationen des ESP8266

- » 802.11 b/g/n
- » Integrierte low power 32-bit MCU
- » Integrierte 10-bit ADC
- » Integrierte TCP/IP protocol stack
- » Integrierte TR switch, balun, LNA, Leistungsverstärker und Anpassungsnetzwerk
- » Integrierte PLL, Regler und Power-Management-Einheiten
- » Unterstützt vielfältige Antennen
- » Wi-Fi 2.4 GHz, unterstützt WPA/WPA2
- » Unterstützt STA/AP/STA+AP Betriebsarten
- » Unterstützung der Smart-Link-Funktion für Android- und iOS-Geräte
- » SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
- » STBC, 1x1 MIMO, 2x1 MIMO
- » A-MPDU & A-MSDU Aggregation und 0,4s Schutzintervall
- » Deep sleep power <10uA, Ableitstrom beim Abschalten < 5uA
- » Wake up und übertragen Pakete in < 2ms
- » Standby Leistungsaufnahme von < 1.0mW (DTIM3)
- » +20dBm Ausgangsleistung in 802.11b mode
- » Betriebstemperaturbereich: -40 °C ~ 125 °C

Das D1 Mini Modul

Das D1-Mini-Modul wird ungelötet mit einem Paar achtpoliger Stiftleisten, einem Paar achtpoliger Buchsenleisten und einem Paar achtpoliger Buchsenleisten mit extra langen Beinen (die auf dem Titelbild zu sehen sind) geliefert.

Spezifikationen

- » Betriebsspannung: 3.3V DC
- » Chip: ESP8266
- » Taktfrequenz: 80MHz / 160MHz
- » Flash: 4MB
- » Digitale I/O pins: 11
- » Analoge eingangs pins: 1
- » Analoge Eingangsspannungsbereich: from 0V to 3.3V DC
- » Port: microUSB
- » USB chip: CH340 chip
- » On-board LED: connected to GPIO2 pin
- » Abmessungen: 25 x 35 x 6mm [0.98 x 1.4 x 0.24in]
- » Max. Strom pro individuellem digitalen I/O-Pin: 12mA

Digitale I/O pins

Wie jede Arduino-Platine hat das D1-Mini-Modul digitale Eingangs-/Ausgangspins oder GPIO - General Purpose Input/Output Pins. Wie der Name schon sagt, können sie als digitale Eingänge verwendet werden, um eine digitale Spannung zu lesen, oder als digitale Ausgänge, um entweder 0 V (Senkenstrom) oder 3,3 V (Quellenstrom) auszugeben.

Das D1 Mini-Modul verfügt über einen Mikrocontroller, der im Spannungsbereich von 0V - 3,3V arbeitet.

Der maximale Strom, der von einem einzelnen GPIO-Pin gezogen werden kann, beträgt 12mA!

Hinweis: Die Pins des D1-Mini-Moduls sind nicht 5V-tolerant, das Anlegen von mehr als 3,6V an einem Pin kann den Chip beschädigen!

GPIO1 und GPIO3 werden als TX und RX der seriellen Hardwareschnittstelle (UART) verwendet, so dass sie in den meisten Fällen nicht als normale E/A beim Senden/Empfangen von seriellen Daten verwendet werden können.

Das D1 Mini-Modul hat eine eingebaute LED, die an den GPIO2-Pin angeschlossen ist.

PWM

Im Gegensatz zu den meisten Atmel-Chips (Arduino) unterstützt das D1 Mini-Modul keine Hardware-PWM. Allerdings wird Software-PWM an allen digitalen Pins unterstützt. Der Standard-PWM-Bereich beträgt 10 Bit bei 1 kHz, kann aber geändert werden (bis zu 14 Bit bei 1 kHz).

Analoger Eingang

Das D1-Mini-Modul verfügt über einen einzelnen Analogeingang, mit einem Eingangsspannungsbereich von 0V - 3,0V. Wenn mehr als 3,3 V angelegt werden, kann der Chip beschädigt werden. Der Analog-Digital-Wandler (ADC) hat eine Auflösung von 10 Bit.

Seriell

Das D1-Mini-Modul hat zwei Hardware-UARTs (serielle Schnittstellen): UART0 an den Pins 1 und 3 (TX0 bzw. RX0), und UART1 an den Pins 2 und 8 (TX1 bzw. RX1). GPIO8 wird jedoch für den Anschluss des Flash-Chips verwendet. Das bedeutet, dass UART1 nur Daten senden kann. In den meisten Fällen ist nur ein UART-Port mehr als genug.

UART0 hat auch eine Hardware-Flusskontrolle an den Pins 15 und 13 (RTS0 bzw. CTS0). Diese beiden Pins können auch als Alternative für TX0- und RX0-Pins verwendet werden.

I2C

Das D1-Mini-Modul verfügt nicht über ein Hardware-I2C oder TWI (Two Wire Interface), sondern ist in Software implementiert. Das bedeutet, dass zwei beliebige digitale Pins als I2C-Pins verwendet werden können. Standardmäßig verwendet die I2C-Bibliothek GPIO4 als SDA und GPIO5 als SCL (im Datenblatt ist GPIO2 als SDA und GPIO14 als SCL spezifiziert). Die maximale Geschwindigkeit des I2C-Taktes beträgt ca. 450kHz.

SPI

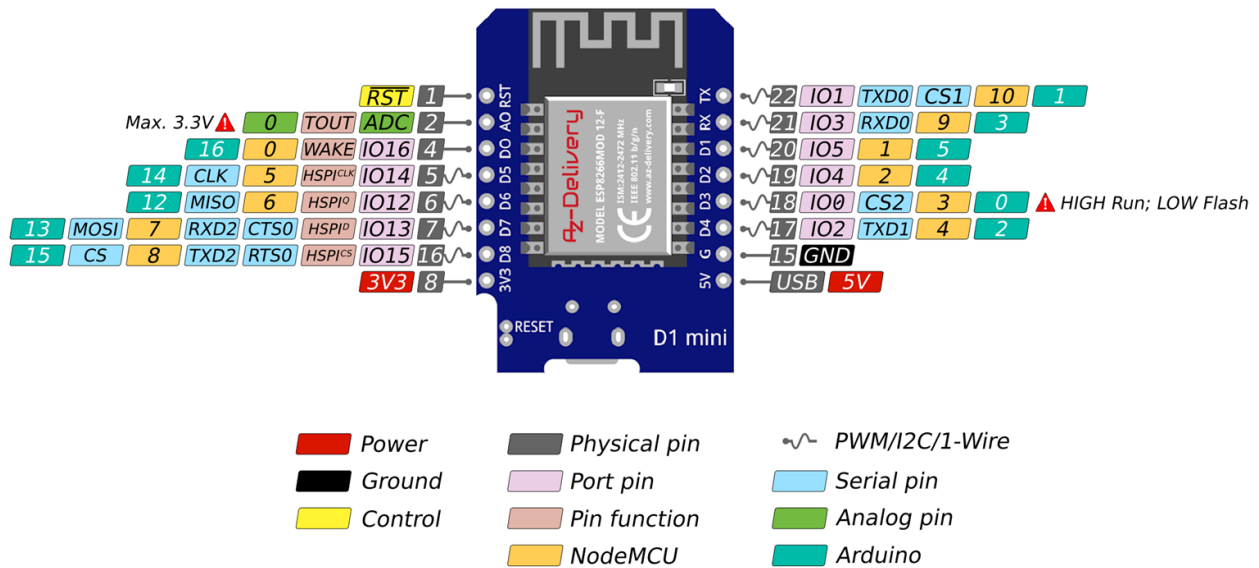
Das D1-Mini-Modul stellt dem Anwender einen SPI-Anschluss zur Verfügung, der als HSPI bezeichnet wird. Er kann sowohl im Slave- als auch im Master-Modus (in Software!) verwendet werden.

Es verwendet:

- *GPIO14* als clock – *CLK*,
- *GPIO12* als *MISO*,
- *GPIO13* als *MOSI* and
- *GPIO15* als Slave Select - *SS*.

Pinbelegung

Das D1-Mini-Modul hat zwei Reihen mit je acht Pins (insgesamt sechzehn Pins). Die Pinbelegung des Moduls ist in der folgenden Abbildung dargestellt:



D1 Mini Modul - Software

Der größte Teil der Mikrocontroller-Funktionalität des ESP verwendet genau die gleiche Syntax wie ein normales Arduino-Board, was den Einstieg sehr einfach macht.

Digitale I/O pins

Genau wie bei einem normalen Arduino-Board kann die Pin-Funktion mit der folgenden Codezeile eingestellt werden:

```
pinMode(pin, mode)
```

wobei pin der Name des GPIO-Pins ist und mode entweder INPUT (was die Voreinstellung ist), oder OUTPUT, oder INPUT_PULLUP sein kann, um die eingebauten Pull-up-Widerstände für die Pins GPIO0 - 15 zu aktivieren. Um den Pull-Down-Widerstand für GPIO16 zu aktivieren, verwenden Sie *INPUT_PULLDOWN_16*.

Um einen Ausgangspin HIGH (3,3 V) oder LOW (0 V) zu setzen, verwenden Sie die folgende Codezeile:

```
digitalWrite(pin, value)
```

wobei pin der Name des GPIO-Pins ist, und der Wert entweder 1 oder 0 (oder HIGH und LOW).

Um einen Eingang zu lesen, verwenden Sie die folgende Codezeile:
`digitalRead(pin)`

Um PWM an einem bestimmten Pin zu aktivieren, verwenden Sie die folgende Codezeile:

```
analogWrite(pin, value)
```

wobei pin der Name des GPIO-Pins ist, und value eine Zahl zwischen 0 und 1023.

Der Bereich des PWM-Ausgangs kann mit der folgenden Code-Zeile geändert werden: `analogWriteRange(new_range)`

Die Frequenz der PWM kann mit der folgenden Code-Zeile geändert werden:

```
analogWriteFreq(new_frequency)
```

wobei new_frequency zwischen 100Hz und 1000Hz liegen sollte.

Analoger Eingang

Wie auf jedem Arduino-Board kann die Funktion `analogRead(A0)` verwendet werden, um die analoge Spannung am Analogeingang zu erhalten (0 = 0V, 1023 = 1,0V).

Das D1-Mini-Modul kann auch den ADC verwenden, um die Versorgungsspannung (VCC) zu messen. Fügen Sie dazu oben in Ihrem Sketch `ADC_MODE(ADC_VCC)` ein und verwenden Sie `ESP.getVcc()`, um die Spannung tatsächlich zu ermitteln.

Hinweis: Wenn ein Analogeingangspin zum Lesen der Versorgungsspannung verwendet wird, kann an den Analogeingangspin nichts anderes angeschlossen werden!

Serielle Kommunikation

Um UART0 (TX = GPIO1, RX = GPIO3) zu verwenden, benutzen Sie das Serial-Objekt, genau wie auf einem Arduino-Board: `Serial.begin(baud_rate)`

Um die alternativen Pins (TX = GPIO15, RX = GPIO13) zu verwenden, verwenden Sie die folgende Codezeile: `Serial.swap()` nach `Serial.begin()`

Um UART1 (TX = GPIO2) zu verwenden, benutzen Sie das Objekt `Serial1`.

Hinweis: Alle Arduino-Stream-Funktionen, wie `read()`, `write()`, `print()`, `println()`, etc. werden ebenfalls unterstützt.

I2C und SPI Schnittstelle

Für die I2C- und SPI-Schnittstelle verwenden Sie die Standard-Syntax der Arduino-Bibliothek.

Gemeinsame Nutzung der CPU-Zeit mit dem RF-Teil

Eine Sache, die Sie beim Schreiben von Programmen für das D1-Mini-Modul (ESP8266) beachten sollten, ist, dass der Sketch Ressourcen (CPU-Zeit und Speicher) mit den WLAN- und TCP-Stacks (der Software, die im Hintergrund läuft und alle WLAN- und IP-Verbindungen verarbeitet) teilen muss. Wenn die Ausführung des Codes zu lange dauert und die TCP-Stacks ihre Arbeit nicht machen können, kann das Programm abstürzen oder die Daten können verloren gehen. Es ist am besten, die Ausführungszeit der Schleife unter ein paar hundert Millisekunden zu halten. Jedes Mal, wenn die Hauptschleife wiederholt wird, gibt ein Sketch an das Wifi und TCP ab, um alle Wifi und TCP-Anfragen zu bearbeiten. Wenn die Schleife länger dauert, muss die CPU-Zeit explizit an die wifi/TCP-Stacks gegeben werden, indem man `delay(0)` oder `yield()` einbezieht. Wenn dies nicht geschieht, funktioniert die Netzwerkkommunikation nicht wie erwartet, und wenn sie länger als 3 Sekunden dauert, setzt der weiche WDT (Watchdog Timer) den ESP zurück. Wenn der Soft-WDT deaktiviert ist, setzt nach etwas mehr als 8 Sekunden der Hardware-WDT den Chip zurück. Aus der Perspektive des Mikrocontrollers sind 3 Sekunden jedoch eine sehr lange Zeit (240 Millionen Taktzyklen), so dass die Skizze davon nicht betroffen ist, es sei denn, es werden extrem schwere Zahlenberechnungen durchgeführt oder extrem lange Strings über die serielle Schnittstelle gesendet. Denken Sie einfach daran, die `yield()` innerhalb der `for`- oder `while`-Schleifen hinzuzufügen, die länger als, sagen wir, 100ms dauern könnten.

Einrichten der Arduino-IDE

Falls die Arduino-IDE nicht installiert ist, folgen Sie dem [link](#) und laden Sie die Installationsdatei für das Betriebssystem Ihrer Wahl herunter.

Download the Arduino IDE

Für



The screenshot shows the Arduino IDE download page. On the left, there is a large teal circle containing the Arduino logo (an infinity symbol with a minus and a plus sign). To the right of the logo, the text reads: **ARDUINO 1.8.9**, followed by a description: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software." Below this, it says: "This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions." On the right side of the page, there is a teal sidebar with links for different operating systems: **Windows** (Installer, for Windows XP and up; Windows ZIP file for non admin install), **Windows app** (Requires Win 8.1 or 10, with a 'Get' button), **Mac OS X** (10.8 Mountain Lion or newer), **Linux** (32 bits, 64 bits, ARM 32 bits, ARM 64 bits), **Release Notes**, **Source Code**, and **Checksums (sha512)**.

Windows Benutzer: Doppelklicken Sie auf die heruntergeladene `.exe`-Datei und folgen Sie den Anweisungen im Installationsfenster.

Für *Linux* Benutzer, laden Sie eine Datei mit der Erweiterung *.tar.xz* herunter, die extrahiert werden muss. Wenn sie extrahiert ist, gehen Sie in das extrahierte Verzeichnis und öffnen Sie das Terminal in diesem Verzeichnis. Zwei *.sh* Skripte müssen ausgeführt werden, das erste namens *arduino-linux-setup.sh* und das zweite heißt *install.sh*.

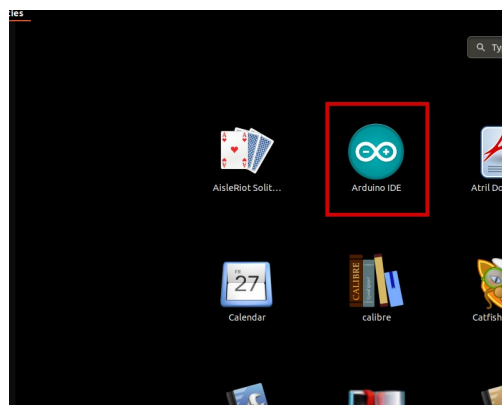
Um das erste Skript im Terminal auszuführen, öffnen Sie das Terminal im extrahierten Ordner und führen Sie den folgenden Befehl aus:

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

user_name - ist der Name eines Superusers im Linux-Betriebssystem. Ein Passwort für den Superuser muss beim Start des Befehls eingegeben werden. Warten Sie einige Minuten, bis das Skript vollständig abgeschlossen ist.

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

Nach der Installation dieser Skripte gehen Sie zu *All Apps*, wo die *Arduino-IDE* installiert ist.

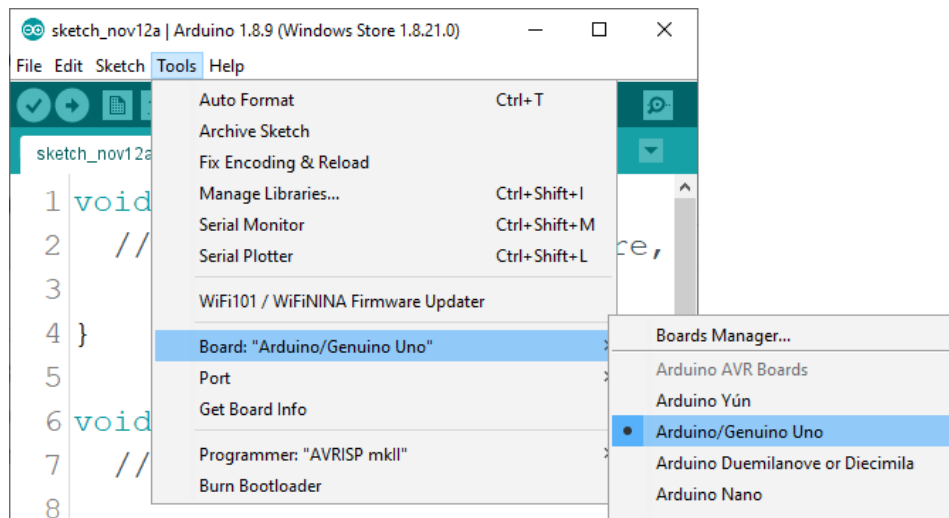


Fast alle Betriebssysteme werden mit einem vorinstallierten Texteditor ausgeliefert (z.B. *Windows* mit *Notepad*, *Linux* Ubuntu mit *Gedit*, *Linux Raspbian* mit *Leafpad* usw.). Alle diese Texteditoren sind für den Zweck des eBooks vollkommen in Ordnung.

Zunächst ist zu prüfen, ob Ihr PC ein Arduino-Board erkennen kann. Öffnen Sie die frisch installierte Arduino-IDE, und gehen Sie zu:

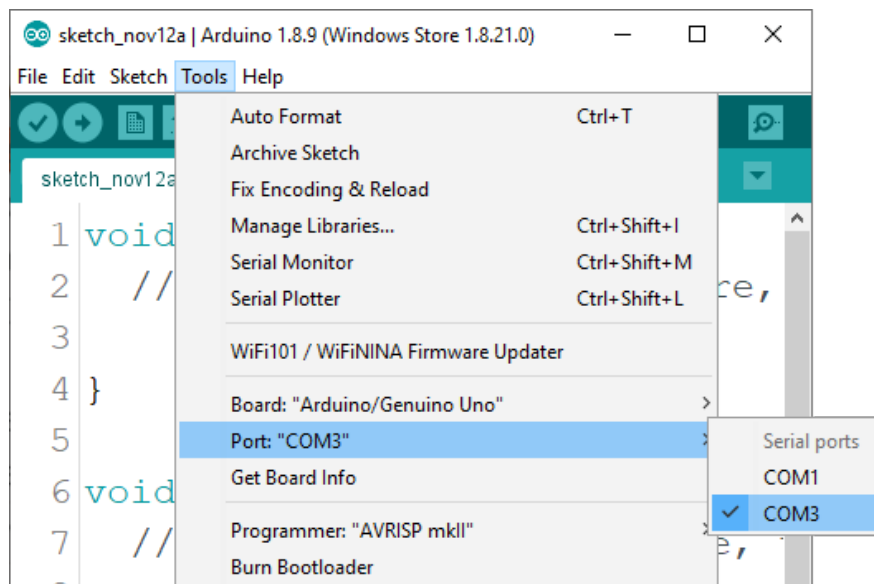
Tools > Board > {your board name here}

{your board name here} sollte der *Arduino/Genuino Uno* sein, wie es auf dem folgenden Bild zu sehen kann:



Der Port, an den das Arduino-Board angeschlossen ist, muss ausgewählt werden. Gehe zu: *Tools > Port > {port name goes here}* und wenn das Arduino-Board an den USB-Port angeschlossen ist, ist der Portname im Drop-down Menü auf dem vorherigen Bild zu sehen.

Wenn die Arduino-IDE unter Windows verwendet wird, lauten die Portnamen wie folgt:



Für *Linux* Benutzer, ist zum Beispiel der Portname `/dev/ttyUSBx`, wobei x für eine ganze Zahl zwischen 0 und 9 steht.

D1 Mini mit Arduino IDE

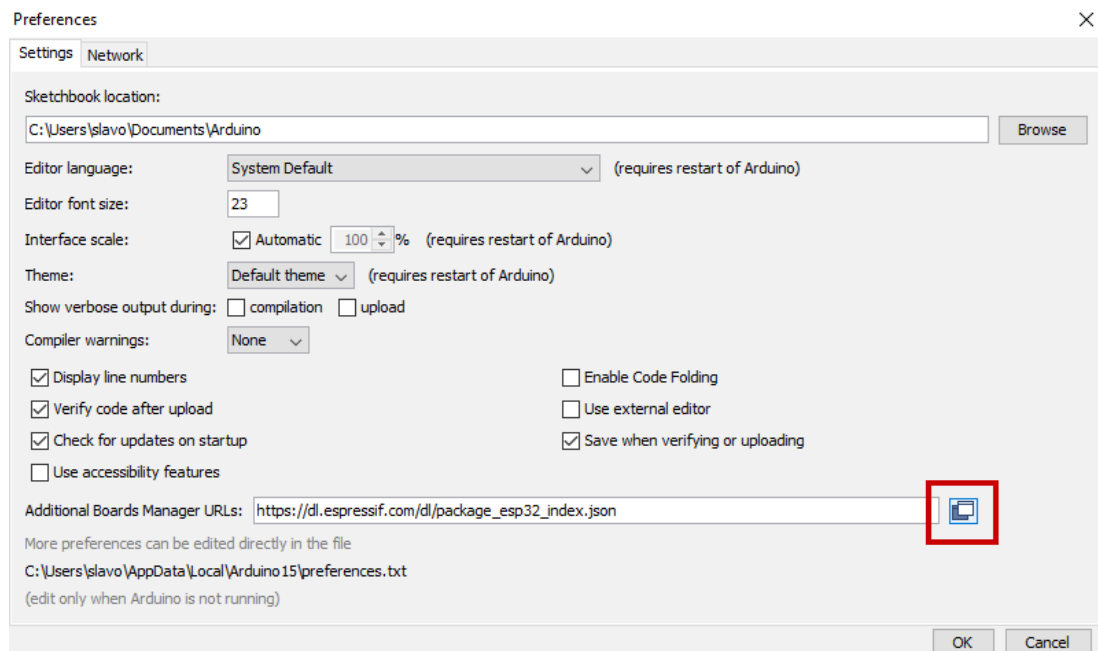
Um die Arduino IDE so einzurichten, dass das D1 Mini darüber programmiert werden kann, folgen Sie ein paar einfachen Schritten.

Als erstes muss der USB Treiber installiert werden, dieser kann [hier](#) heruntergeladen werden.

Danach müssen Sie den ESP8266-Core installieren. Um ihn zu installieren, öffnen Sie die Arduino IDE und gehen Sie zu:

File > Preferences

und finden Sie das Feld *Additional URLs*.



Dann kopieren Sie folgende URL:

https://arduino.esp8266.com/stable/package_esp8266com_index.json

und fügen Sie ihn in das Feld *Additional URLs* ein.

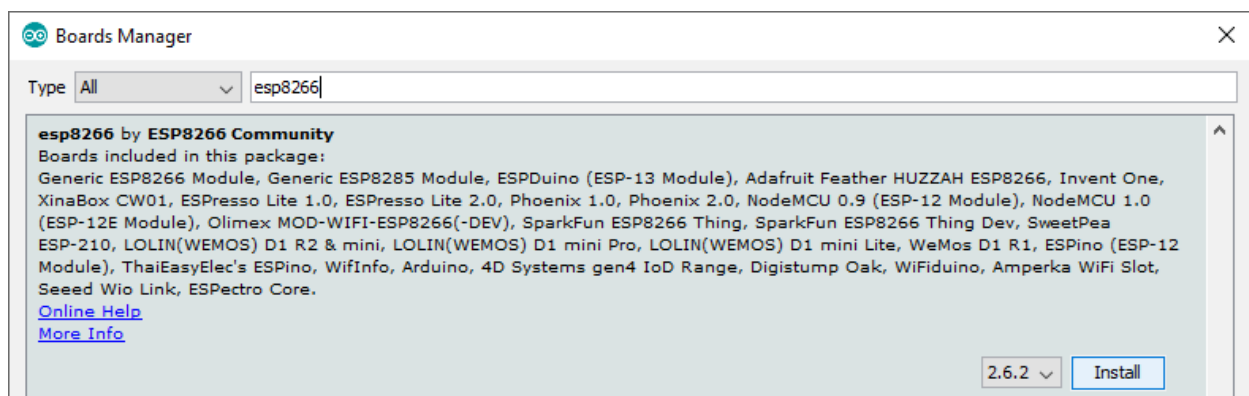
Wenn Sie bereits einen oder mehrere Links in diesem Feld haben, fügen Sie einfach ein Komma nach dem letzten Link ein, fügen Sie den neuen Link nach dem Komma ein und klicken Sie auf die Schaltfläche **OK**. Schließen Sie dann die Arduino-IDE.



Öffnen Sie die Arduino IDE und gehen Sie zu:

Tools > Board > Boards Manager

Es öffnet sich ein neues Fenster, geben Sie "esp8266" in das Suchfeld ein und installieren Sie das Board mit dem Namen "esp8266" von "ESP8266 Community", wie unten abgebildet:

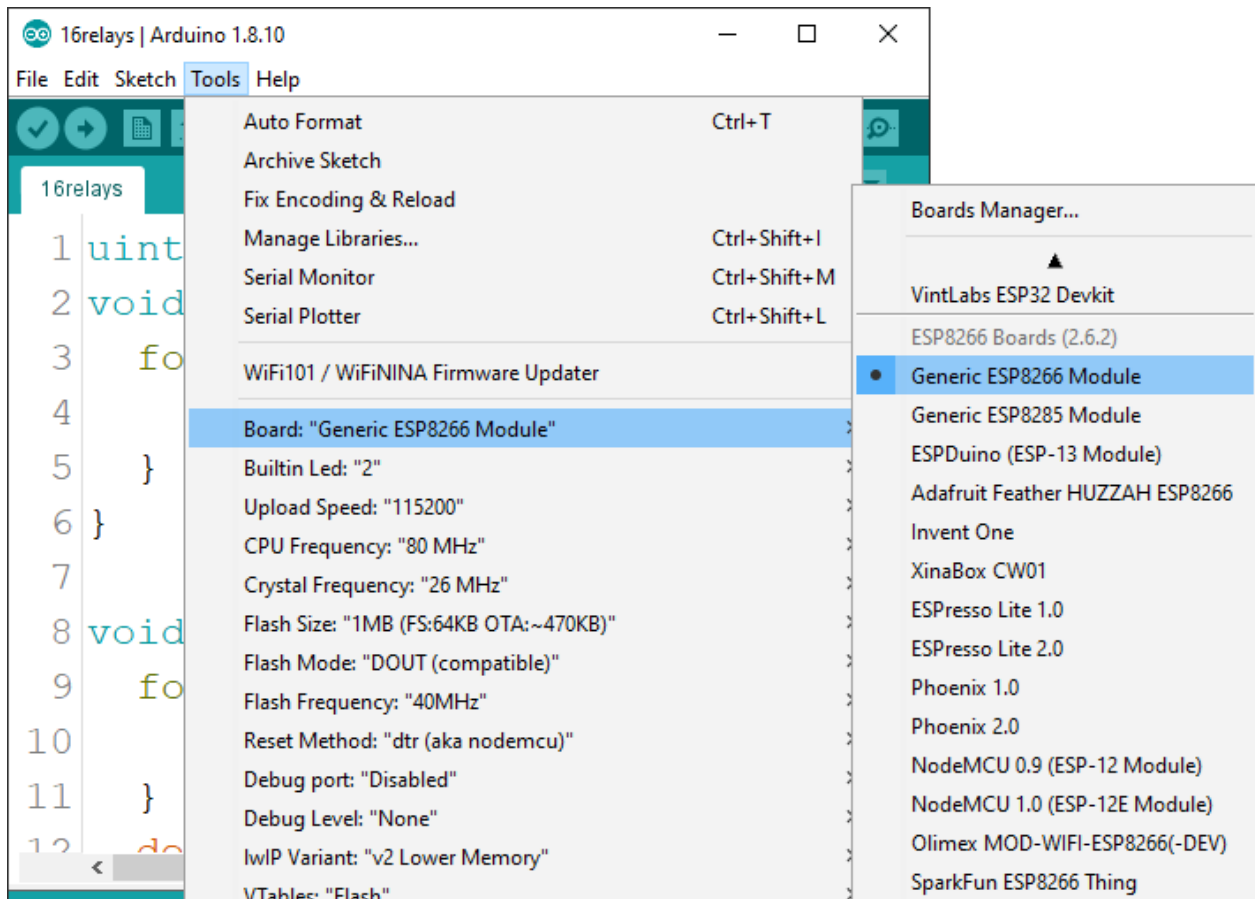


Jetzt haben Sie den ESP8266-Core installiert.

Als Nächstes wählen Sie das richtige Board in der Arduino IDE aus. Öffnen Sie die Arduino IDE und gehen Sie zu:

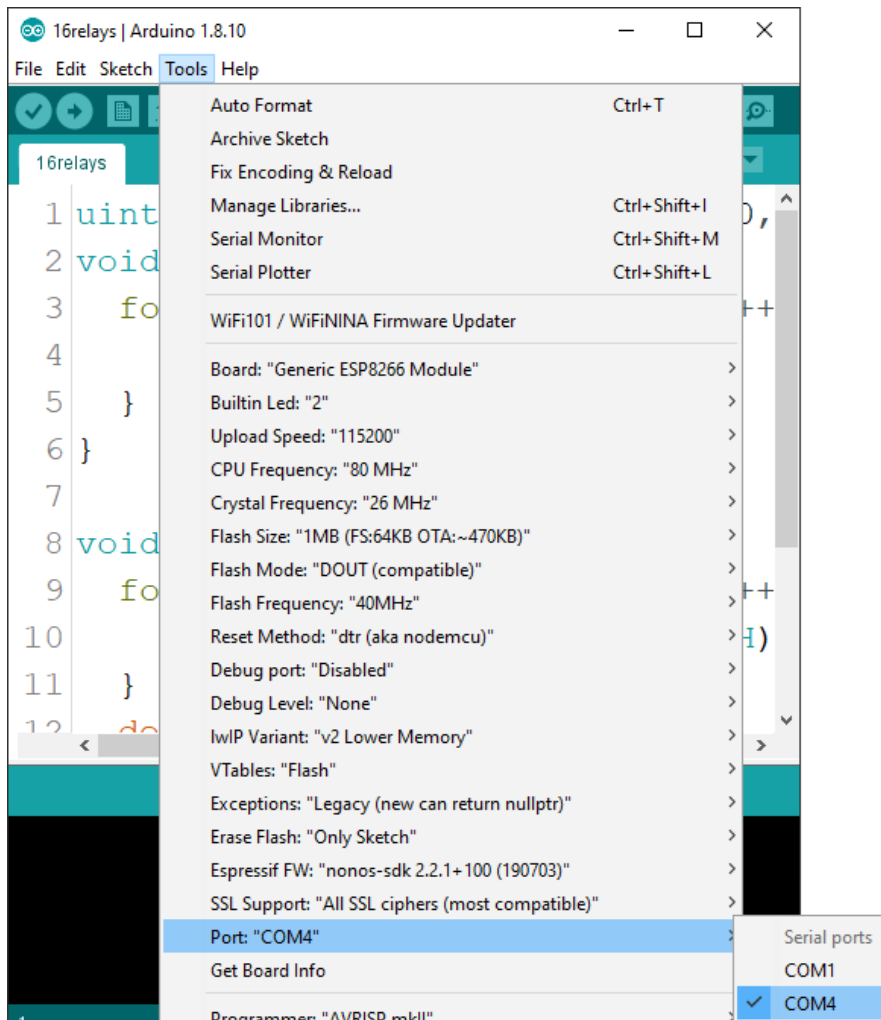
Tools > Board > {board name}

und wählen Sie das erste *Generic ESP8266 Module* aus, wie unten abgebildet:



Wählen Sie danach den Port aus, an dem das D1 Mini verbunden ist. Gehen Sie zu: *Tools > Port > {port name goes here}*

Wenn das D1 Mini Modul v3 über den USB-Port verbunden ist, sollten mehrere Portnamen verfügbar sein. In diesem eBook wird die *Arduino IDE* in *Windows* verwendet, die Portnamen lauten daher wie folgt:



For Linux users, port name is */dev/ttyUSBx* for example, where “x” represents specific integer number between 0 and 9.

Blink + PWM + Serial Beispiel Sketche

Blink Beispiel Sketch

Es gibt ein Blink-Sketch-Beispiel, das mit der ESP8266-Board-Bibliothek geliefert wird. Um es zu öffnen, gehen Sie zu: *Datei > Beispiele > ESP8266 > Blink*

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);    // Initialize the LED_BUILTIN  
}  
void loop() {  
    digitalWrite(LED_BUILTIN, LOW);  // Turn the LED on  
    // Note that LOW is the voltage level  
    // but actually the LED is on; this is because  
    // it is active LOW on the ESP  
    delay(1000);                      // Wait for a second  
    digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off  
    // by making the voltage HIGH  
    delay(2000);                      // Wait for two seconds  
}
```

Beim D1-Mini-Modul ist LED_BUILTIN gleich einer Zahl 2, was bedeutet, dass die On-Board-LED mit dem GPIO2-Pin verbunden ist. Um die LED einzuschalten, muss der GPIO2-Pin in den LOW-Zustand gebracht werden, und um sie auszuschalten, muss der GPIO2-Pin in den HIGH-Zustand gebracht werden.

Software PWM Beispiel Sketch

```
int brightness = 1; // do not set it to the zero
                        // zero disables the PWM on a specific pin
uint8_t fadeAmount = 5;
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
    analogWrite(LED_BUILTIN, 200); // high brightness
    delay(1000);
    analogWrite(LED_BUILTIN, 500);
    delay(1000);
    analogWrite(LED_BUILTIN, 800);
    delay(1000);
    analogWrite(LED_BUILTIN, 1000); // low brightness
    delay(1000);

    // fading led
    while (1) {
        analogWrite(LED_BUILTIN, brightness);
        brightness = brightness + fadeAmount;
        if (brightness < 0 || brightness >= 1023) {
            fadeAmount = -fadeAmount;
        }
        delay(10);
    }
}
```

Beim D1-Mini-Modul ist LED_BUILTIN gleich einer Zahl 2, was bedeutet, dass die On-Board-LED mit dem GPIO2-Pin verbunden ist. Um die LED einzuschalten, muss der GPIO2-Pin in den LOW-Zustand gebracht werden, und um sie auszuschalten, muss der GPIO2-Pin in den HIGH-Zustand gebracht werden.

Verwenden Sie für den Wert nicht die Zahl Null, da dadurch die PWM-Funktion an einem bestimmten Pin ausgeschaltet wird!

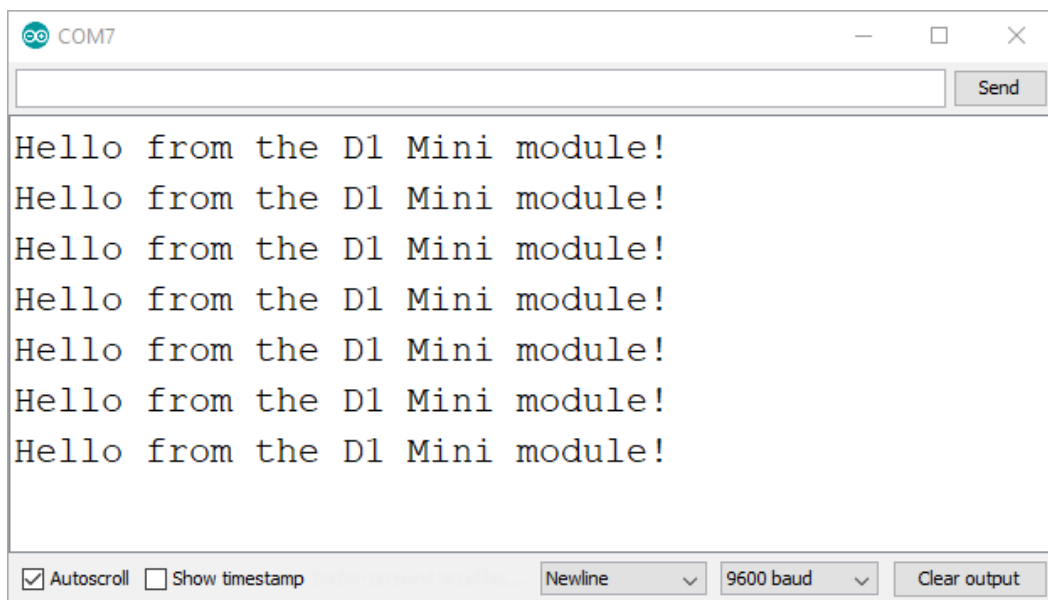
Näher am Wert Null ist die höhere Helligkeitsstufe einer LED und näher am Wert 1023 ist die niedrigere Helligkeitsstufe einer LED.

Um die LED zum Ausblenden zu bringen, muss die while(1)-Schleife innerhalb der loop()-Funktion verwendet werden, sonst funktioniert es nicht.

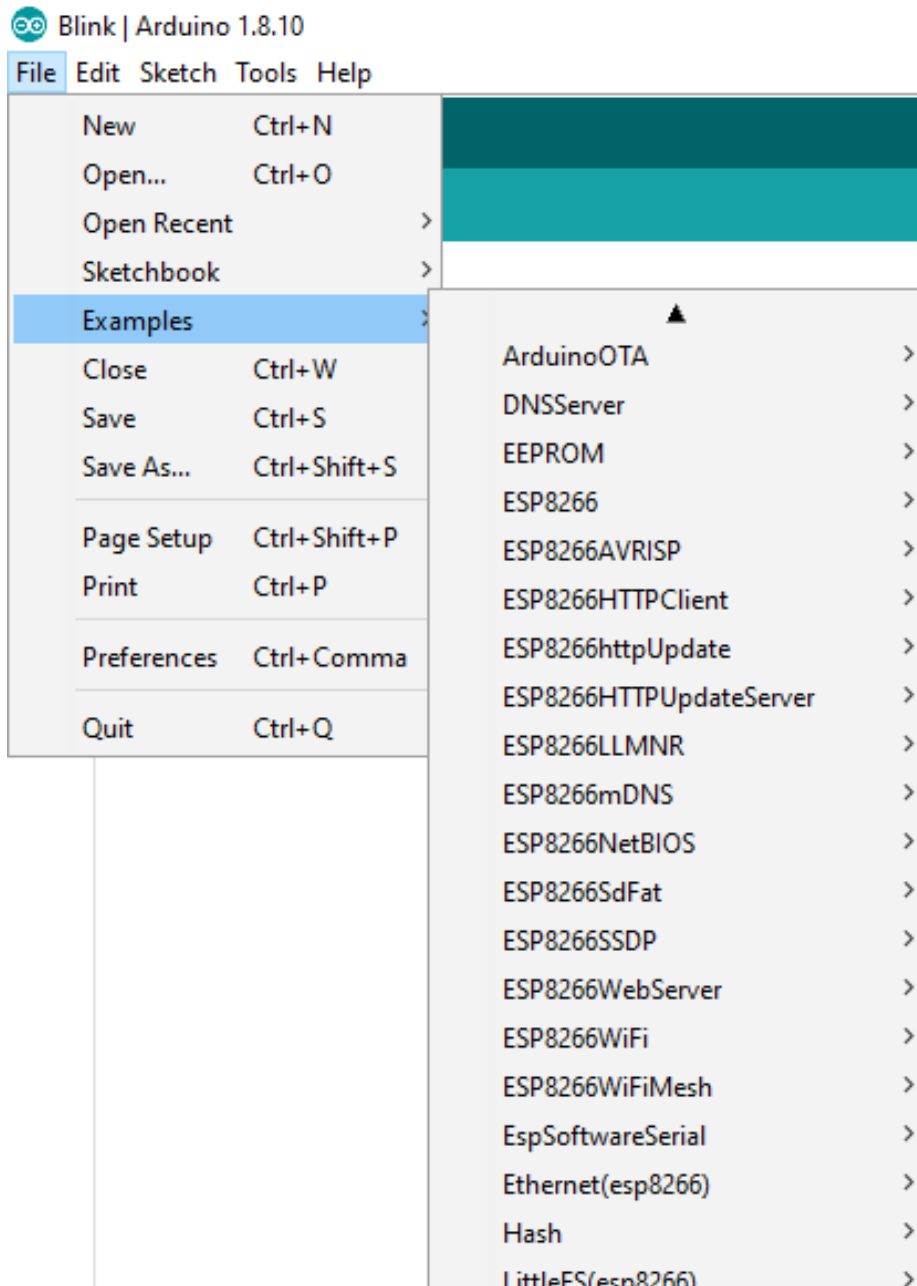
Serielle Kommunikation Beispiel Sketch

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("Hello from the D1 Mini module!");  
  delay(1000);  
}
```

Laden Sie den Sketch auf das D1-Mini-Modul und öffnen Sie den Serial Monitor (Werkzeuge > Serieller Monitor). Das Ergebnis sollte wie auf dem folgenden Bild aussehen:



Mit der ESP8266-Board-Bibliothek kommen viele weitere Sketch-Beispiele. Der WiFi-Teil des D1-Mini-Moduls könnte von dort aus getestet werden. Er wird in diesem eBook nicht behandelt.



Jetzt sind Sie dran! Entwickeln Sie Ihre eigenen Projekte und Smart-Home Installationen. Wie Sie das bewerkstelligen können, zeigen wir Ihnen unkompliziert und verständlich auf unserem Blog. Dort bieten wir Ihnen Beispielskripte und Tutorials mit interessanten kleinen Projekten an, um schnell in die Welt der Mikroelektronik einzusteigen. Zusätzlich bietet Ihnen auch das Internet unzählige Möglichkeiten, um sich in Sachen Mikroelektronik weiterzubilden.

Falls Sie nach weiteren hochwertigen Produkten für Arduino und Raspberry Pi suchen, sind Sie bei AZ-Delivery Vertriebs GmbH goldrichtig. Wir bieten Ihnen zahlreiche Anwendungsbeispiele, ausführliche Installationsanleitungen, E-Books, Bibliotheken und natürlich die Unterstützung unserer technischen Experten.

<https://az-delivery.de>

Viel Spaß!

Impressum

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