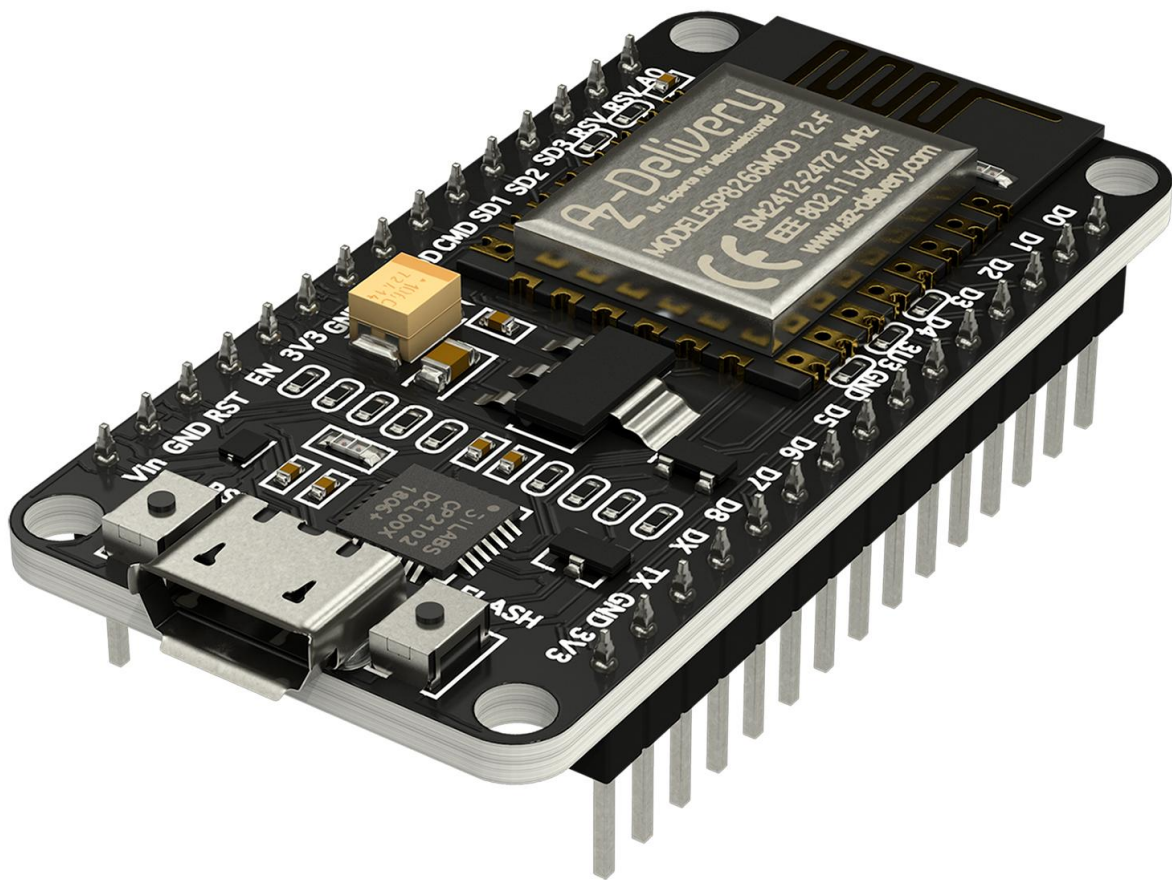


NodeMCU Amica V2

E-Book

NodeMCU Amica V2



Einführung	3
Spezifikationen	4
NodeMCU	6
Spezifikationen von NodeMCU	7
NodeMCU LUA Amica V2 Pinout	9
Boot-Modi	11
Interne Pull-Up/Down-Widerstände	13
Digitale E/A-Pins	13
PWM - Pulsweitenmodulation	14
Analoger Input	15
WiFi-Kommunikation	17
Weitere Spezifikationen	17
PWM - Impulsbreitenmodulation	21

Einführung

NodeMCU LUA Amica V2 ist ein Entwicklungsboard rund um den ESP8266-Chip. Es enthält einen Spannungsregler und eine USB-Programmierschaltung für den ESP8266-Chip sowie einige andere Funktionen.

Für die Entwicklung von Anwendungen können Sie zwischen der Arduino IDE und der Sprache LUA wählen. Die Maker-Community ist sehr aktiv und unterstützt Plattformen wie ESP8266.

NodeMCU wird mit einer vorinstallierten Firmware geliefert, die es uns ermöglicht, mit der interpretierten Sprache LUA zu arbeiten und Befehle über den seriellen Anschluss (CP2102-Chip) zu senden. Das NodeMCU-Board ist eine der meistgenutzten Plattformen für Internet of Things (IoT)-Projekte. NodeMCU LUA Amica V2 ist voll kompatibel mit der Arduino IDE.

Die NodeMCU-Platine ist speziell für die Arbeit auf dem Breadboard konzipiert. Es hat einen Plattenspannungsregler, der es ermöglicht, direkt vom USB-Anschluss zu speisen. Die Eingangs-/Ausgangsstifte arbeiten mit 3,3 V. Der CP2102-Chip ist für die USB-zu-Seriell-Kommunikation zuständig.

Spezifikationen

Versorgungsspannung (USB):	5V DC
Eingangs-/Ausgangsspannung:	3.3V DC
SoC:	ESP8266 (ESP-12 Modul)
CPU:	Tensilica Xtensa LX3 (32 Bit)
Taktfrequenz:	80MHz / 160MHz
Anleitung RAM:	32kB
Daten-RAM:	96kB
Externer Flash-Speicher:	4MB
Digitale GPIO-Pins:	17 (kann als PWM bei 3,3V konfiguriert werden)
ADC-Analog-Pin:	1 (BUT Spannungsbereich ist: 0 - 1V)
UART:	2
Serieller USB-Chip	CP2102
Integriert	TCP/IP-Protokoll-Stack
Ausgangsleistung von	+ 19,5 dBm im 802.11b-Modus
Ableitstrom	weniger als 10uA
Aufwachen und Übertragen von Paketen in	<2ms
Leistungsaufnahme im Standby-Modus	<1.0mW (DTIM3)
Wi-Fi-Direkt (P2P), Soft-AP	
PCB-Antenne	



Die ESP8266-Serie von Wi-Fi-Chips wird von Espressif Systemen, einem Halbleiterunternehmen aus Shanghai (China), hergestellt. Der ESP8266 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, UART und mehr. Das Beste daran ist, dass es mit einem drahtlosen Netzwerk ausgestattet ist, was es von anderen Mikrocontrollern wie dem Atmega328p unterscheidet. Das bedeutet, dass Sie zu einem erschwinglichen Preis Geräte einfach über Wi-Fi fernsteuern und überwachen können.

ESP8266 ist ein System-on-Chip (SoC), der einen 32-Bit Tensilica-Mikrocontroller, digitale Standard-Peripherieschnittstellen, Antennenschalter, RF-Balun, Leistungsverstärker, rauscharmen Empfangsverstärker, Filter und Power-Management-Modul in einem kleinen Gehäuse integriert. Er bietet 2.4GHz Wi-Fi (802.11 b/g/n, unterstützt sowohl WPA als auch WPA2), 17 GPIO-Pins, I2C (IIC)-Schnittstelle, Analog-Digital-Wandlung (10-bit, auf einem Pin), SPI-Schnittstelle, UART (auf dedizierten Pins, plus ein reiner Sende-UART kann auf GPIO2 aktiviert werden) und PWM (Pulsweitenmodulation) in Software auf jedem GPIO-Pin.

Der Prozessorkern, der von Espressif "L106" genannt wird, basiert auf dem Tensilica Diamond Standard 106Micro 32-Bit-Prozessorcontrollerkern und läuft mit 80MHz. Er verfügt über ein 64kB Boot-ROM, 32kB Befehls-RAM und 80kB Benutzerdaten-RAM. Auf den externen Flash-Speicher kann über die SPI-Schnittstelle zugegriffen werden.

Die Hersteller haben daher viele kompakte Leiterplattenmodule entwickelt, die auf dem ESP8266-Chip basieren (wie NodeMCU LUA Amica V2). Einige dieser Module haben spezifische Bezeichnungen, wie "ESP-01" bis "ESP-14". ESP8266-basierte Module erweisen sich als kostengünstige, netzwerkfähige und vielseitig einsetzbare Plattform zur Erleichterung der IoT-Entwicklung am Endpunkt.

NodeMCU

Die NodeMCU ist ein Open-Source-Firmware- und Entwicklungskit, das Ihnen hilft, Ihr IoT-Produkt mit wenigen Lua-Skriptzeilen zu prototypisieren.

NodeMCU ist eine quelloffene IoT-Plattform. Sie umfasst Firmware, die auf dem ESP8266 läuft, und Hardware, die auf dem ESP-12-Modul basiert. Der Begriff "NodeMCU" bezieht sich auf die Firmware und nicht auf die Entwicklungskits.

Die Firmware verwendet die Skriptsprache Lua. Sie basiert auf dem eLua-Projekt und baut auf dem Espressif Non-OS SDK für ESP8266 auf.

Spezifikationen von NodeMCU

- Entwicklungskit - basiert auf ESP8266
- Atmega328p-ähnliche Hardware-I
- Netzwerk-API im Nodejs-Stil - Ereignisgesteuerte API für Netzwerk Anwendungen
- Kostengünstigstes Wi-Fi

Unterschied zwischen ESP8266 (NodeMCU) und Atmega328p

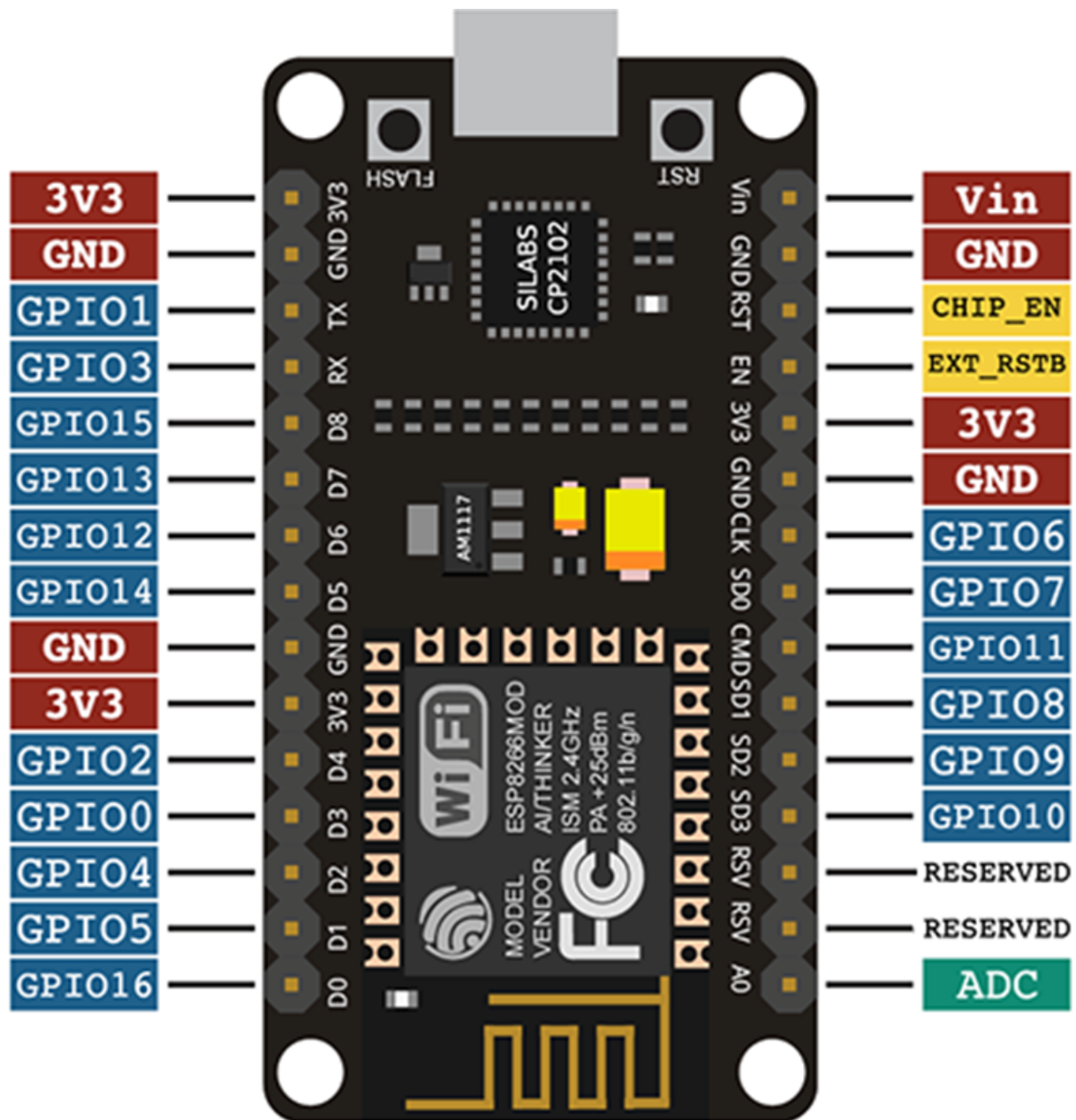
Spezifikation	MC	ESP8266
RAM:	4kB	80kB
FLASH-Speicher:	32kB	4MB
Geschwindigkeit:	16MHz	80MHz
GPIOs (nutzbar):	14	11
E/A-Spannungspegel:	5V	3.3V
ADC (Auflösung):	6 (10-Bit)	1 (10-Bit)
Serielle Schnittstelle	1	1
I2C-Schnittstelle:	1	1
SPI-Schnittstelle:	1	vom Flash-Chip verwendet
PWM, Auflösung:	6, 8 Bit	Alle GPIO Pins, 10 Bit
WiFi	Nein	Ja 2MBps

Unterschied zwischen ESP8266 and ESP32

Spezifikation	ESP8266	ESP32
MCU:	Xtensa Einzelkern 32-Bit L106	Xtensa Doppelkern 32-Bit LX6
802.11 b/g/n Wi-Fi:	HT20	HT40
Bluetooth:	Nein	Bluetooth 4.2 und BLE
Typische Frequenz	80MHz	160MHz
SRAM:	Nein	Ja
Blitzlicht:	Nein	Ja
GPIO Pins:	17	36
HW/SW PWM:	Keine / 8 Kanäle	Keine / 16 Kanäle
SPI/I2C/I2S/UART:	2/1/2/2	4/2/2/2
ADC:	10-Bit	12-Bit
CAN:	Nein	Ja
Ethernet-MAC-Schnitt stelle	Nein	Ja
Berührungssensor:	Nein	Ja
Temperatursensor:	Nein	Ja
Hall-Effekt-Sensor:	Nein	Ja
Arbeitstemperatur:	von -40°C bis 125°C	von -40°C bis 125°C

NodeMCU LUA Amica V2

Pinout



GPIO Pin-Beschreibung

Genau wie ein normales Atmega328p-Board hat der ESP8266 digitale Eingangs-/Ausgangs-Pins (GPIO-Pins - Allzweck-Eingangs-/Ausgangs-Pins). Diese digitalen Eingänge/Ausgänge arbeiten mit 3,3V.

5V Spannung sollte nicht an die ESP8266 Chip Pins angeschlossen werden!

Die Pins sind nicht 5V-tolerant, die Anwendung von mehr als 3,6V auf einem Pin wird den Chip zerstören.

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

Der ESP8266 hat 17 GPIO-Pins, von denen Sie jedoch nur 11 nutzen können, da 6 Pins (GPIO 6 - 11) für den Anschluss des Flash-Speicherchips verwendet werden. Dies ist der kleine 8-Pin IC direkt neben dem ESP8266. Wenn Sie versuchen, einen dieser Pins zu verwenden, könnte Ihr Programm abstürzen.

GPIO 1 und 3 werden als TX und RX der seriellen Hardware-Schnittstelle (UART) verwendet, so dass sie in den meisten Fällen nicht als normale E/A beim Senden/Empfangen von seriellen Daten verwendet werden können.

Boot-Modi

Nur wenige E/A-Pins haben während des Bootens eine besondere Funktion: Sie wählen einen von drei Boot-Modi aus:

GPIO15	GPIO0	GPIO2	Modus
0V	0V	3.3V	UART Bootloader
0V	3.3V	3.3V	Boot-Sketch (SPI flash)
3.3V	X	X	SDIO-Modus (für Atmega328p nicht verwendet)

HINWEIS: Ein externer Pull-Down-Widerstand von $1k\Omega$ ist für GPIO0 erforderlich, ein externer Pull-Up-Widerstand für GPIO2 ist nicht erforderlich, da der interne Pull-Up-Widerstand an diesem Pin beim Booten aktiviert ist.

- GPIO15 ist immer auf NIEDRIG gezogen, so dass Sie den internen Pull-up-Widerstand an diesem Pin nicht verwenden können. Beachten Sie dies, wenn Sie GPIO15 als Eingang zum Lesen eines Schalters verwenden oder ihn an ein Gerät mit einem Open-Collector- (oder Open-Drain-) Ausgang wie I2C anschließen.

- GPIO0 wird im Normalbetrieb auf HOCH gezogen.
- GPIO2 kann beim Booten nicht auf NIEDRIG sein, daher kann man keinen Schalter daran anschließen.

Interne Pull-Up/Down-Widerstände

GPIO 0-15 haben alle einen eingebauten Pull-Up-Widerstand, genau wie bei einem Atmega328p. GPIO16 hat einen eingebauten Pull-Down-Widerstand.

Digitale E/A-Pins

Sie können die Funktion eines Pins einstellen mit:

einem Pin-Modus (*Pin*, *Modus*), wobei *Pin* die GPIO-Nummer ist und *Modus* kann entweder INPUT sein, was die Voreinstellung ist, OUTPUT oder INPUT_PULLUP, um die eingebauten Pull-Up-Widerstände für GPIO 0 - 15 zu aktivieren. Um den Pull-Down-Widerstand für GPIO16 zu aktivieren, verwenden Sie INPUT_PULLDOWN_16.

wobei *Pin* die GPIO-Nummer ist, und *Modus* kann entweder INPUT sein, was die Voreinstellung ist, OUTPUT oder INPUT_PULLUP, um die eingebauten Pull-Up-Widerstände für GPIO 0 - 15 zu aktivieren. Um den Pull-Down-Widerstand für GPIO16 zu aktivieren, verwenden Sie INPUT_PULLDOWN_16.

Um einen Ausgangspin auf HOCH (3,3 V) oder NIEDRIG (0 V) zu setzen, verwenden Sie:

digitalWrite (*Pin*, *Wert*), wobei *Pin* der digitale Anschluss ist und *Wert* entweder 1 oder 0 (oder HOCH und NIEDRIG).

Um einen Eingang zu lesen, verwenden Sie digitalRead(*pin*) .

PWM - Pulsweitenmodulation

Der ESP8266 unterstützt Software-PWM auf allen digitalen Pins. Die Standard-PWM-Auflösung ist 10-Bit bei 1kHz, kann aber geändert werden. Um PWM an einem bestimmten Pin zu aktivieren, verwenden Sie:

`analogWrite (Pin, Wert)`

wobei *Pin* der digitale Anschluss ist

und den *Wert* eine Zahl zwischen 0 und 1023.

Sie können den Bereich (Bittiefe) des PWM-Ausgangs ändern, indem Sie *analogWriteRange (range)* verwenden.

Die Frequenz kann mit *analogWriteFreq (frequency)* geändert werden.

Die Frequenz sollte zwischen 100Hz und 1000Hz liegen.

Analoger Input

Der ESP8266 hat einen einzigen analogen Eingangspin mit einem Eingangsspannungsbereich von 0,0V bis 1,0V. Wenn Sie z.B. eine Spannung von 3,3V liefern, wird der Chip beschädigt. Der NodeMCU verfügt über einen integrierten ohmschen Spannungsteiler, um einen einfacheren Bereich von 0 - 3,3V zu erhalten. Der A/D-Wandler (Analog-Digital-Wandler) hat eine Auflösung von 10 Bit. Der ESP kann den A/D-Wandler auch nutzen, um die Versorgungsspannung (VCC) zu messen. Dazu müssen Sie Folgendes einbeziehen:

`A/D-Wandler_MODUS (A/D-Wandler_VCC)`

am Anfang Ihres Sketches und verwenden Sie:

`ESP.getVcc()`

um tatsächlich die Spannung zu erhalten. Wenn Sie es verwenden, um die Versorgungsspannung zu lesen, können Sie nichts anderes an den Analog-Pin anschließen.

Serielle Kommunikation

Der ESP8266 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), allerdings wird GPIO8 für den Anschluss des Flash-Chips verwendet. Das bedeutet, dass UART1 nur Daten übertragen kann.

Zusätzlich verfügt UART0 über eine Hardware-Flusskontrolle an den Pins 15 und 13 (RTS0 bzw. CTS0). Diese beiden Pins können auch als alternative TX0- und RX0-Pins verwendet werden.

Um UART0 (TX = GPIO1, RX = GPIO3) zu verwenden, können Sie das Serial-Objekt verwenden, genau wie bei einem Atmega328p: `Serial.begin(baud)`

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

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

WiFi-Kommunikation

ESP kann in drei verschiedenen Modi arbeiten: Wi-Fi-Station, Wi-Fi-Zugangspunkt und beides gleichzeitig.

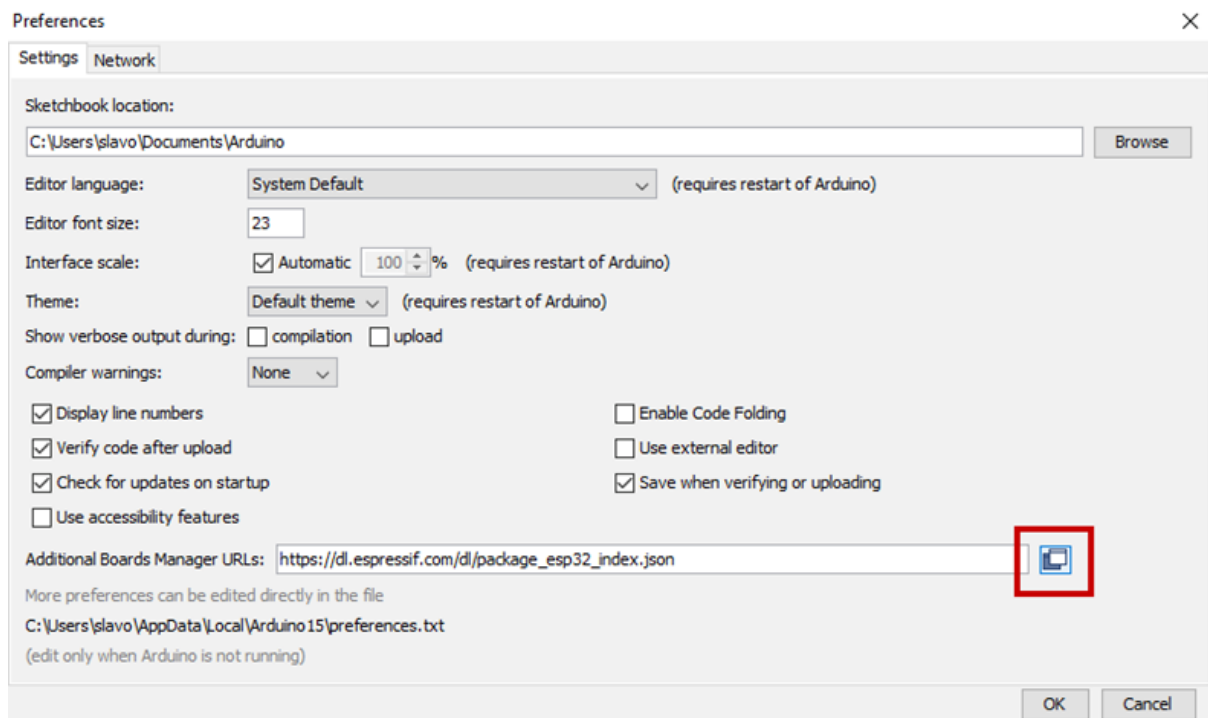
Weitere Spezifikationen

Eines der Spezifikationen von NodeMCU LUA Amica V2 ist seine Fähigkeit, als Access Point oder Hotspot für Ihr Wi-Fi-Projekt zu arbeiten. Sie können auch einen Webserver auf dem NodeMCU laufen lassen. Eine weitere Funktion ermöglicht das Hochladen von Code auf NodeMCU LUA Amica V2 über das Internet. Dies wird als OTA - Over-The-Air-Programmierung bezeichnet und ist ein Prozess, der es Geräten ermöglicht, ihre Firmware oder Software drahtlos ohne physischen Zugang zu aktualisieren (über Wi-Fi, Bluetooth, GPRS oder 4G/3G). Diese Funktionen werden in diesem E-Book nicht behandelt.

Einrichtung des ESP8266/NodeMCU mit Arduino IDE

Um NodeMCU mit Arduino IDE zu verwenden, folgen Sie ein paar einfachen Schritten. Zuerst müssen Sie den ESP8266-Kern installieren. Um ihn zu installieren, öffnen Sie Arduino IDE und wählen Sie aus:

File > Preferences, und finden Sie *Additional URLs* Feld.



Kopieren Sie dann die folgende URL:

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

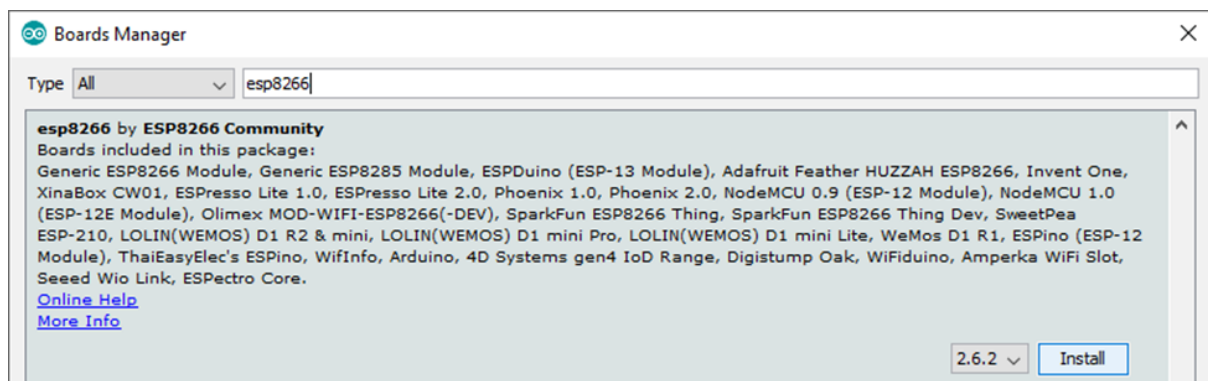
Fügen Sie diesen Link in das Feld Additional URLs ein. Wenn Sie bereits einen oder mehrere Links in diesem Feld haben, fügen Sie einfach ein Komma nach dem letzten Link ein, dann fügen Sie den neuen Link nach dem Komma ein und klicken Sie auf die OK-Taste. Schließen Sie dann die Arduino IDE.



Öffnen Sie die Arduino IDE erneut und wählen Sie aus:

Tools > Board > Boards Manager

Es öffnet sich ein neues Fenster. Geben Sie "esp8266" in das Suchfeld ein und installieren Sie das Board "esp8266" von der "ESP8266 Community", wie auf dem Bild unten zu sehen:



Nun haben Sie den ESP8266-Kern installiert.

Um das NodeMCU LUA Amica V2 Board auszuwählen, wählen Sie aus:

Tools > Board > NodeMCU 1.0 (ESP - 12E Module)

Um den Sketch-Code auf das NodeMCU-Board hochzuladen, wählen Sie zunächst den Port aus, an den Sie das Board angeschlossen haben. Dann die Option:

Tools > Port > {port name}

Blinken zweier LEDs auf dem NodeMCU LUA Amica V2

Es gibt zwei LEDs auf dem NodeMCU LUA Amica V2. Eine LED ist mit dem GPIO-Pin 2 verbunden und befindet sich auf dem ESP8266-Board. Die andere LED ist mit dem GPIO-Pin 16 verbunden und befindet sich auf dem NodeMCU-Board. Wenn Sie das standardmäßige Blink-Sketch-Beispiel verwenden, das mit der Arduino IDE geliefert wird, stellt das LED_BUILTIN-Makro die LED dar, die mit dem GPIO-Pin 2 verbunden ist. Um diese beiden LEDs zum Blinken zu bringen, ist der folgende Code das Sketch-Beispiel:

```
#define led_built_in_ESP 2
#define led_built_in_Node 16
void setup() {
  pinMode(led_built_in_ESP, OUTPUT);
```

```
pinMode(led_built_in_Node, OUTPUT);
}
void loop() {
digitalWrite(led_built_in_ESP, HIGH);
digitalWrite(led_built_in_Node, LOW);
delay(1000);
digitalWrite(led_built_in_ESP, LOW);
digitalWrite(led_built_in_Node, HIGH);
}
```

PWM-

Impulsbreitenmodulation

Wir werden PWM verwenden, um eine an GPIO-Pin 2 angeschlossene LED auszublenden. Sketch-Code:

```
#define LED 2
uint16_t brightness = 0; // how bright the LED is
uint8_t fadeAmount = 5; // how many points to fade the
LED by
void setup() {
  pinMode(LED, OUTPUT);
}
void loop() {
  analogWrite(LED, 0);
  delay(2000);
  analogWrite(LED, 512);
  delay(2000);
}
```

```
analogWrite(LED, 1023);  
delay(2000);  
while(1) {  
    analogWrite(LED, brightness);  
    brightness = brightness + fadeAmount;  
    if(brightness <= 0 || brightness >= 1023) {  
        fadeAmount = -fadeAmount;  
    }  
    delay(15);  
}
```

Sie haben es geschafft!

**Jetzt können Sie Ihr Modul für verschiedene Projekte
verwenden.**

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

Wenn Sie auf der Suche nach hochwertigen Arduino- und Raspberry-Pi-kompatiblen Peripheriegeräten sind, sind Sie bei der AZ-Delivery Vertriebs GmbH genau an der richtigen Adresse. Sie erhalten zahlreiche Anwendungsbeispiele, vollständige Installationsanleitungen, E-Books, Bibliotheken und Unterstützung durch unsere technischen Experten.

<https://az-delivery.de>

Viel Spaß!

Impressum

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