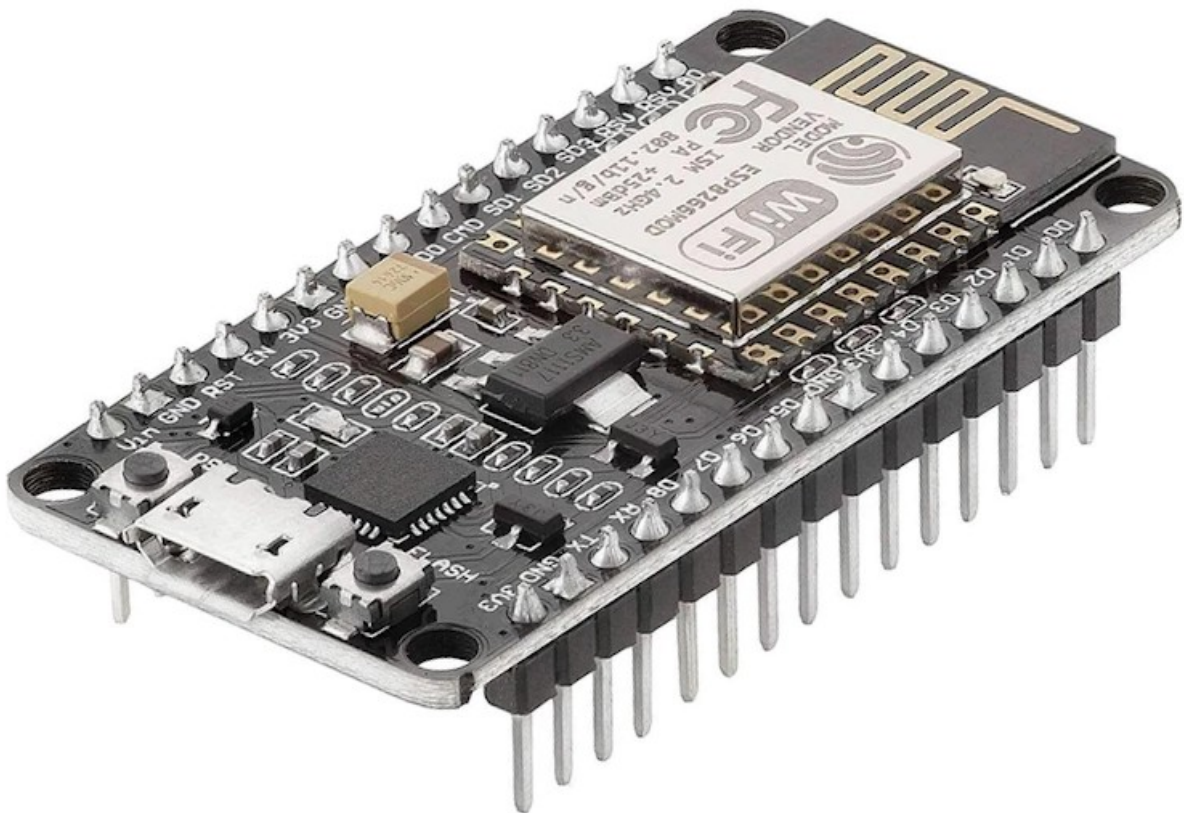


AZ-Delivery

Willkommen!

Vielen Dank, dass sie sich für unser *NodeMCU LUA Amica V2* von *AZ-Delivery* entschieden haben. In den nachfolgenden Seiten werden wir Ihnen erklären wie Sie das Gerät einrichten und nutzen können.

Viel Spaß!





NodeMCU LUA Amica V2 ist ein Entwicklungsboard, das um den *ESP8266*-Chip herum entwickelt wurde und einen Spannungsregler und eine USB-Programmierschaltung für den *ESP8266*-Chip sowie einige andere Funktionen enthält.

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

NodeMCU wird mit einer vorinstallierten Firmware geliefert, die es uns ermöglicht, mit der interpretierten Skriptsprache LUA zu arbeiten, durch das Senden von Befehlen über die serielle Schnittstelle (CP2102-Chip). Das NodeMCU-Board ist eine der meistgenutzten Plattformen für "Internet of Things" (IoT) Projekte. NodeMCU LUA Amica V2 ist vollständig kompatibel mit der Arduino IDE.

Das NodeMCU-Board ist speziell dafür konzipiert, dass es mit dem Breadboard funktioniert. Es verfügt über einen Plattenspannungsregler, der es ermöglicht, es direkt vom USB-Port zu speisen. Die Input/Output Pins arbeiten bei 3,3V. Der CP2102-Chip ist für die USB-Seriell-Kommunikation verantwortlich.



Technische Daten:

- » Versorgungsspannung (USB): 5V DC
- » Input/Output Spannung: 3.3V DC
- » SoC: ESP8266 (ESP-12 Modul)
- » CPU: Tensilica Xtensa LX3 (32 bit)
- » Taktfrequenz: 80MHz / 160MHz
- » Instruction RAM: 32kB
- » Data RAM: 96kB
- » Externer Flashspeicher: 4MB
- » GPIO digitale Pins: 17 (als PWM bei 3,3V konfigurierbar)
- » ADC analoge Pin: 1 (ABER Spannungsbereich ist: 0 - 1V)
- » UART: 2
- » USB Seriell-Chip: CP2102
- » PCB Antenne
- » 802.11 b/g/n
- » Wi-Fi Direct (P2P), soft-AP
- » Integrierter TCP/IP-Protokollstapel
- » Ausgangsleistung von + 19,5dBm im 802.11b-Modus
- » Leckstrom weniger als 10uA
- » Wake up und Übertragungspakete in <2ms
- » Standby-Stromverbrauch <1.0mW (DTIM3)

Az-Delivery



Die ESP8266-Serie von Wi-Fi-Chips wird von Espressif Systems, einem Halbleiterunternehmen aus Shanghai (China), hergestellt. ESP8266 ist ein erschwingliches Wi-Fi-Modul, das sich für Heimwerkerprojekte im Bereich "Internet of Things" (IoT) eignet. Dieses Modul wird mit vielen GPIOs und Unterstützung für eine Vielzahl von Protokollen wie SPI, I2C, UART und mehr geliefert. Das Beste daran ist, dass es mit einer drahtlosen Vernetzung ausgestattet ist, wodurch es sich von anderen Mikrocontrollern wie dem Atmega328p unterscheidet. Das bedeutet, dass Sie Geräte einfach und kostengünstig über Wi-Fi fernsteuern und überwachen können.

ESP8266 ist ein System-on-Chip (SoC), der einen 32-Bit-Tensilica-Mikrocontroller, digitale Standard-Peripherieschnittstellen, Antennenschalter, HF-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 sowohl WPA als auch WPA2), 17 GPIO-Pins, I2C (IIC)-Schnittstelle, Analog-Digital-Wandler (10-Bit, auf einem Pin), SPI-Schnittstelle, UART (auf zugeordneten Pins, zudem kann ein "transmit-only" UART auf GPIO2 aktiviert werden) und Software-PWM (Pulsweitenmodulation) auf jedem GPIO-Pin.

Az-Delivery

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

Die Hersteller haben daher viele kompakte Leiterplattenmodule auf der Basis des ESP8266-Chips entwickelt (wie NodeMCU LUA Amica V2). Einige dieser Module haben spezifische Kennungen, wie z.B. "*ESP-01*" bis "*ESP-14*". ESP8266-basierte Module haben sich als kostengünstige, netzwerkfähige und allgemeine Mehrzweckplattform zur Erleichterung der Endpunkt-IoT-Entwicklung erwiesen.



NodeMCU

Die NodeMCU ist ein Open-Source-Firmware- und Entwicklungskit, das Ihnen hilft, Ihr IoT-Produkt innerhalb weniger Lua-Skriptzeilen aufzubauen. Die NodeMCU ist eine Open-Source-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.

Features der NodeMCU:

- Entwicklungs-Kit - basierend auf ESP8266
- Atmega328p-ähnliche Hardware IO
- Netzwerk-API im Nodejs-Stil - Ereignisgesteuerte API für Netzwerkanwendungen
- Kostengünstigstes WI-FI



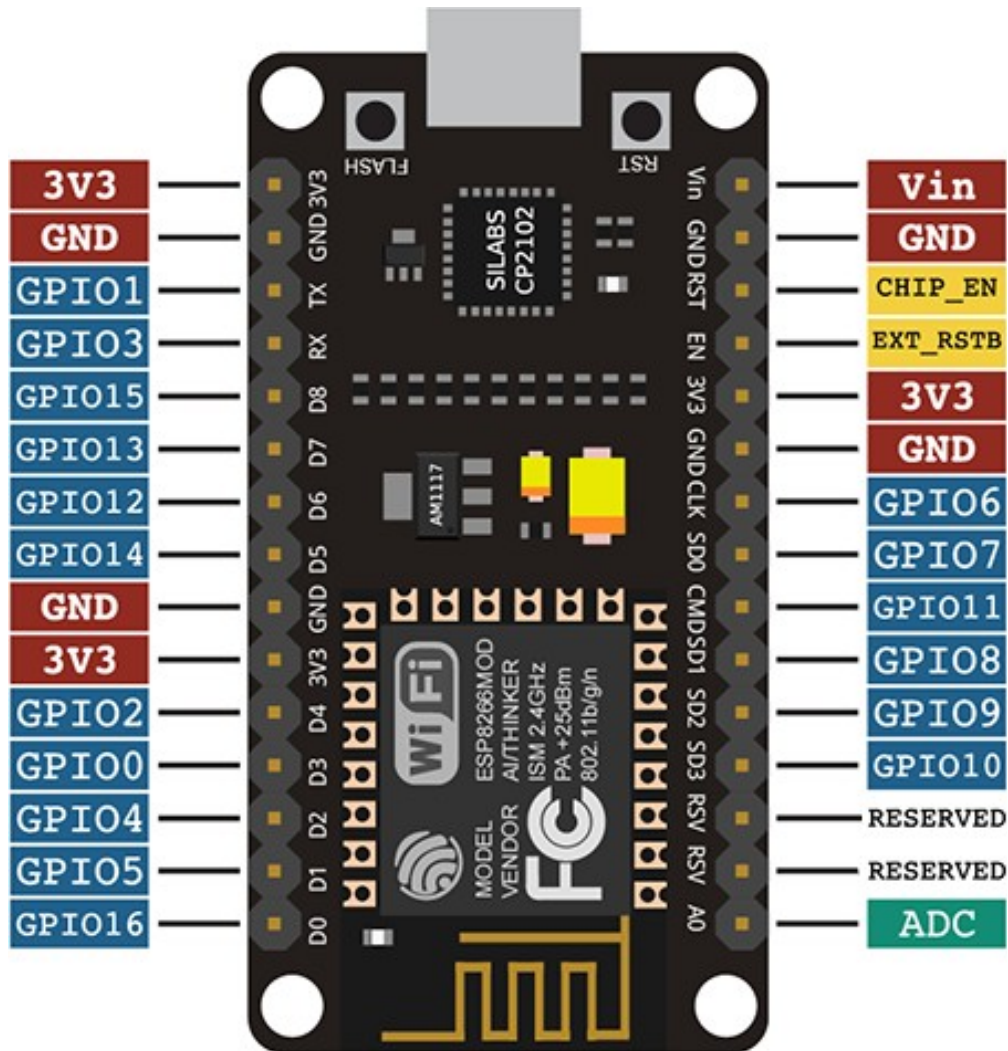
Unterschied zwischen ESP8266 (NodeMCU) und ATmega328p

Spezifikation	Mc	ESP8266
RAM:	4kB	80kB
FLASH-Speicher:	32kB	4MB
Geschwindigkeit:	16MHz	80MHz
GPIOs (verwendbar):	14	11
I/O-Spannungslevel:	5V	3.3V
ADC (Auflösung):	6 (10-bit)	1 (10-Bit)
Serielle Schnittstelle:	1	1
I2C Schnittstelle:	1	1
SPI Schnittstelle:	1	Genutzt von Flash-Chip
PWM, Auflösung:	6, 8 bit	Alle GPIO Pins, 10 bit
WiFi	Nein	Ja 2MBps

Unterschied zwischen ESP8266 und ESP32

Spezifikation	ESP8266	ESP32
MCU:	Xtensa Single-core	Xtensa Dual-Core
	32-bit L106	32-bit LX6
802.11 b/g/n Wi-Fi:	HT20	HT40
Bluetooth:	Nein	Bluetooth 4.2 + BLE
Typische Frequenz:	80MHz	160MHz
SRAM:	Nein	Ja
Flash:	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 Schnittstelle:	Nein	Ja
Touch Sensor:	Nein	Ja
Temperatur Sensor:	Nein	Ja
Hall-Effekt Sensor:	Nein	Ja
Betriebstemperatur:	-40°C bis 125°C	-40°C bis 125°C

NodeMCU LUA Amica V2 Pinbelegung





GPIO Pin Beschreibung

Genau wie ein normales Atmega328p-Board hat das ESP8266 digitale Ein-/Ausgangspins (GPIO-Pins - General Purpose Input/Output Pins). Diese digitalen Ein-/Ausgänge arbeiten mit 3,3V.

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

Die Pins sind nicht 5V-tolerant, das Anlegen von mehr als 3,6V an jedem Pin zerstört den Chip.

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

ESP8266 hat 17 GPIO-Pins, Sie können jedoch nur 11 davon verwenden, 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, könnten Sie Ihr Programm zum Absturz bringen.

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



Boot Modi

Nur wenige I/O-Pins haben während des Bootens eine spezielle Funktion, sie wählen einen von drei Boot-Modi aus:

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

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

- GPIO15 wird immer auf "LOW" gestellt, so dass Sie den internen Pullup-Widerstand an diesem Pin nicht verwenden können. Beachten Sie dies, wenn Sie GPIO15 als Input zum Lesen eines Schalters oder zum Anschluss an ein Gerät mit einem Open-Collector- (oder Open-Drain-) Output, wie z.B. I2C, verwenden.
- GPIO0 wird während des normalen Betriebs auf "HIGH" gestellt.
- GPIO2 kann beim Hochfahren nicht auf "LOW" sein, so dass Sie keinen Schalter daran anschließen können.

Interne Pull-Up/Down-Widerstände



GPIO 0-15 haben alle einen eingebauten Pullup-Widerstand, genau wie bei einem Atmega328p. GPIO16 hat einen eingebauten Pulldown-Widerstand.



Digitale I/O Pins

So können Sie die Funktion eines Pins einstellen:

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

wobei *pin* die GPIO-Nummer ist,

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

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

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

wobei *pin* der digitale Pin,

und *value* entweder 1 oder 0 (oder HIGH und LOW).

Um einen Input zu lesen, verwenden Sie `digitalRead(pin)`.



PWM – Pulsweitenmodulation

ESP8266 unterstützt Software-PWM auf allen digitalen Pins. Die Standard-PWM-Auflösung beträgt 10 Bit bei 1 kHz, aber dies kann geändert werden.

Um die PWM auf einem bestimmten Pin zu aktivieren, verwenden Sie:

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

wobei *pin* der digitale Pin ist,

und *value* eine Zahl zwischen 0 und 1023.

Sie können den Bereich (Bit-Tiefe) des PWM-Ausgangs ändern, indem Sie `analogWriteRange(range)` eingeben.

Die Frequenz kann geändert werden mit

```
analogWriteFreq(frequency)
```

Die Frequenz sollte zwischen 100Hz und 1000Hz liegen.



Analoger Input

Der ESP8266 hat einen einzelnen analogen Eingangspin mit einem Inputspannungsbereich von $0,0V$ bis $1,0V$. Wenn Sie z.B. eine Spannung von $3,3V$ anlegen, beschädigen Sie den Chip. Die NodeMCU hat einen integrierten ohmschen Spannungsteiler, um einen einfacheren Bereich von $0 - 3,3V$ zu erhalten. Der ADC (Analog-Digital-Wandler) hat eine Auflösung von 10 Bit. Der ESP kann den ADC auch zur Messung der Versorgungsspannung (VCC) verwenden. Um dies zu tun fügen Sie:

```
ADC_MODE(ADC_VCC)
```

am Anfang Ihres Sketches bei und verwenden:

```
ESP.getVcc()
```

um die Spannung tatsächlich zu erhalten. Wenn Sie ihn zum Auslesen der Versorgungsspannung verwenden, können Sie nichts anderes an den analogen Pin anschließen.



Serielle Kommunikation

Der ESP8266 hat zwei Hardware-UARTS (serielle Schnittstellen):

- UART0 an den Pins 1 und 3 (jeweils TX0 und RX0), und
- UART1 an den Pins 2 und 8 (jeweils TX1 und RX1), jedoch wird GPIO8 zum 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 (jeweils RTS0 und CTS0). Diese beiden Pins können auch alternativ als TX0- und RX0-Pins verwendet werden.

Um UART0 (TX = GPIO1, RX = GPIO3) zu verwenden, können Sie das Serial-Objekt wie bei einem Atmega328p verwenden:
`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 `Serial1` Objekt.



WiFi Kommunikation

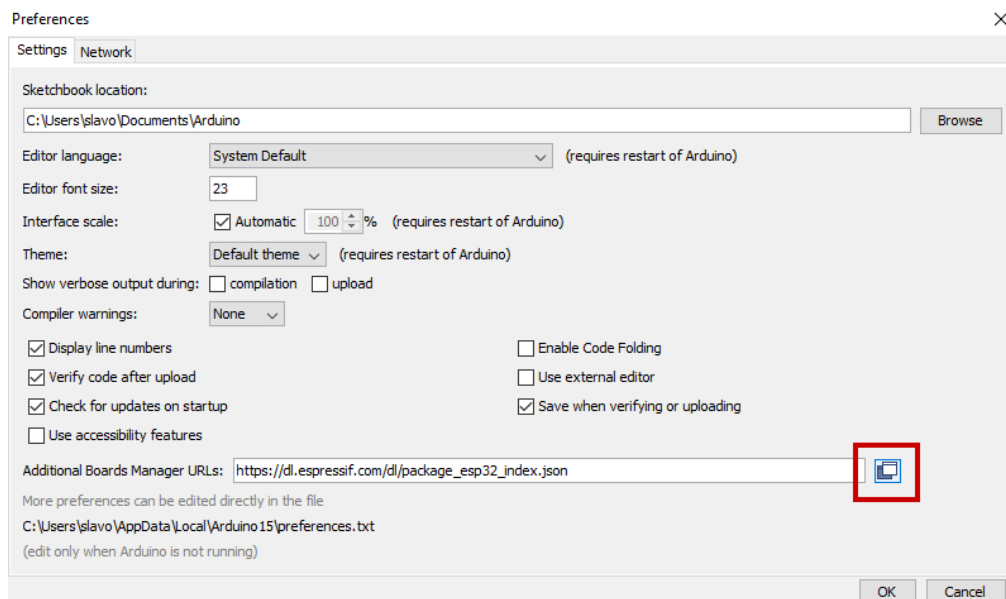
ESP kann in drei verschiedenen Modi arbeiten: Wi-Fi-Station, Wi-Fi-Access Point und beides gleichzeitig.

Sonstige Features

Eines der Features von NodeMCU LUA Amica V2 ist die Fähigkeit, als Access Point oder Hotspot für Ihr Wi-Fi-Projekt zu arbeiten. Außerdem können Sie einen Webserver auf der NodeMCU betreiben. Ein weiteres Feature ermöglicht das Hochladen eines Codes auf NodeMCU LUA Amica V2 über das Internet. Dies wird als OTA - Over-The-Air-Programmierung bezeichnet und ist ein Prozess, der es den Geräten ermöglicht, ihre Firmware oder Software drahtlos und ohne physischen Zugang (über Wi-Fi, Bluetooth, GPRS oder 4G/3G) zu aktualisieren. Diese Funktionen werden in diesem eBook nicht behandelt.

Wie man ESP8266/NodeMCU mit Arduino IDE benutzt

Um NodeMCU mit der Arduino IDE zu verwenden, folgen Sie ein paar einfachen Schritten. Als erstes 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

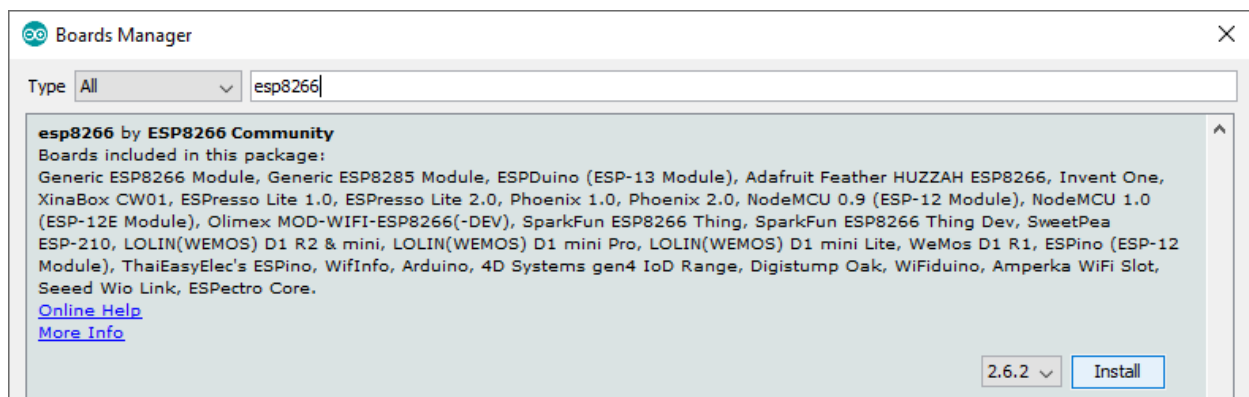
Fügen Sie diesen Link in das Feld "Additional URLs" ein. Wenn Sie bereits einen oder mehrere Links in diesem Feld haben, fügen Sie einfach ein Komma nach dem letzten Link ein, fügen Sie den neuen Link nach dem Komma ein und klicken Sie auf die Schaltfläche OK. Schließen Sie dann die Arduino-IDE.



Öffnen Sie die Arduino IDE erneut 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 auf dem Bild unten gezeigt:



Jetzt haben Sie den ESP8266-Core installiert.

Um das NodeMCU LUA Amica V2-Board auszuwählen, gehen Sie zu:

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

Um den Sketchcode auf die NodeMCU-Karte hochzuladen, wählen Sie zunächst den Port aus, an dem Sie die Karte angeschlossen haben. Gehen Sie zu:

Tools > Port > {port name}



Zwei blinkende LED an Bord der NodeMCU LUA Amica V2

On-Board der NodeMCU LUA Amica V2 befinden sich zwei LEDs. 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 der NodeMCU-Platine. Wenn Sie das standardmäßige *Blink* Sketchbeispiel verwenden, das mit der Arduino-IDE geliefert wird, stellt das "macro" `LED_BUILTIN` die an den GPIO-Pin 2 angeschlossene LED dar: Um diese beiden LEDs zum Blinken zu bringen geht mit folgendem Code als Sketchbeispiel:

```
#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);
    delay(1000);
}
```

```
}
```

PWM – Pulsweitenmodulation

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. Sie können jetzt unser
Modul für Ihre Projekte nutzen.**



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 noch nach weiteren Hochwertige Mikroelektronik und Zubehör suchen, sind Sie bei der AZ-Delivery Vertriebs GmbH goldrichtig. Wir bieten Ihnen zahlreiche Anwendungsbeispiele, ausführliche Installationsanleitungen, Ebooks, Bibliotheken und natürlich die Unterstützung unserer technischen Experten.

<https://az-delivery.de>

Viel Spass!

Impressum

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