

# ESP8266 - erste Erfahrungen

[Originalartikel](#)

[Backup](#)

Im nachfolgenden Artikel möchte ich euch den ESP8266 vorstellen sowie meine ersten Erfahrungen mit diesem interessanten und günstigen Stück Hardware (für weitergehende Informationen habe ich unter dem Artikel eine kleine Linkliste erstellt). Dazu gehe ich erst kurz darauf ein, was der ESP8266 ist, dann werde ich beschreiben, wie man ihn programmieren kann und schließlich zeige ich noch praktische Einsatzzwecke auf. Dazu wird es in diesem Blog in nächster Zeit auch noch ein paar Artikel geben, die diese praktischen Umsetzungen im Detail beschreiben.

Der ESP8266 ist ein WLAN-SoC (System-on-a-Chip), also ein WiFi-Modul, das sich selbstständig nutzen und programmieren lässt. Daher kann der ESP8266 auch unabhängig von z.B. einem Arduino verwendet werden, da man eigene Programme darauf laden kann. Dieser Umstand machte den ESP8266 schnell beliebt in der Bastlerszene, vor allem, da er zu einem sehr günstigen Preis zu bekommen ist (je nach Ausführung für ca. 1-5€ aus China).

Hierbei ist zu erwähnen, dass es mittlerweile sehr viele Ausführungen gibt. Der Chip (also der ESP8266) darauf ist überall identisch, die Boards unterscheiden sich aber an der Anzahl der nutzbaren GPIOs (sprich Anschluss-Pins). Einen Überblick über gängige Module findet ihr z.B. [hier](http://www.esp8266.com/wiki/doku.php?id=esp8266-module-family).



Außerdem gibt es auch noch ein Entwicklungsboard mit dem Namen NodeMCU, das sich bequem über den verbreiteten USB-Anschluss programmieren lässt (bei den anderen Modellen benötigt man zusätzlich einen USB-to-UART-Adapter, wie den FT232RL, den es in China aber bereits für unter 2€ gibt).



Mit 80MHz ist der ESP8266 übrigens schneller getaktet als die meisten Arduinos, die (in den günstigen Modellen) mit 8 bzw. 16 MHz takten. Für einen Microcontroller ist das recht flott; ein weiterer Vorteil zu Gunsten des ESP8266.

Wichtig ist, dass der ESP8266 mit 3V betrieben wird; nicht 5V oder mehr wie bei einigen Arduinos möglich. Eine Stromversorgung direkt über USB ohne Spannungsteiler o.ä. ist somit nicht möglich. Auch haben nicht alle ESP8266-Varianten einen analogen Eingang (nur die -07er [16 Pin Modell] und -12er Modelle); auch hier ein

großer Unterschied zu den Arduinos, die i.d.R. gleich mehrere analogen Eingänge zur Verfügung stellen. Wer also analogen Sensoren (Lichtsensoren, Thermistoren etc.) am ESP8266 betreiben möchte, sollte also aufpassen, welches Modell er sich zulegt. Auch unterscheiden sich die Modelle hinsichtlich ihrer Speicher- (Flash-) Größe (von 0,5MB bis 4MB). Wer größere Programme nutzen möchte, sollte sich vor Kauf erkundigen, wie groß der Speicher auf dem gewünschten Board ist.

Im Gegensatz zum Arduino, der in aller Regel mit der eigens dafür entwickelten Arduino IDE programmiert wird, habt ihr beim ESP8266 mehr Möglichkeiten, eigene Programme auf das Gerät zu bekommen.

Grundsätzlich gibt es 2 Ansätze: Die Nutzung einer bestehenden Firmware, die z.B. Micropython- oder Lua-Scripte entgegennimmt und ausführt oder das Erstellen einer eigenen Firmware z.B. über die Arduino IDE.

Da ich selbst bisher nur die Arduino-IDE verwendet habe, verweise ich an dieser Stelle auf andere Informationsquellen wie den [ESP8266-Artikel auf mikrocontroller.net](https://www.mikrocontroller.net/articles/ESP8266).

Angeschlossen werden die ESP8266-Module wie oben beschrieben über einen USB-to-UART-Adapter. Zu beachten ist hierbei, dass dieser mit 3,3V betrieben werden muss (nicht mit den 5V direkt von USB!). Zudem sollte eine externe Stromversorgung (3,3V) eingesetzt werden, da es Probleme geben kann, wenn der ESP8266 nur durch den Adapter mit Strom versorgt wird. Behelfsweise kann man auch einen Elko zwischen VCC und GND klemmen (habe 220µF genommen), was den ESP8266 wesentlich stabiler laufen lässt.

Zudem sollte man sich vor Inbetriebnahme erkundigen, welche zusätzlichen Pins auf Masse bzw. VCC gelegt werden müssen, um sein jeweiliges Modell programmieren zu können. Beim einfachen, aber beliebten ESP-01 z.B. muss [CH\\_PD auf 3,3V und GPIO\\_0 auf GND gesetzt werden](http://makezine.com/2015/04/01/installing-building-arduino-sketch-5-microcontroller/), um ihn zu programmieren. Bei meinem ESP-201 muss zusätzlich noch [GPIO\\_15 auf GND gelegt werden](http://smarpl.com/content/esp8266-esp-201-module-first-impressions).

Meinen ESP-201 habe ich also so mit dem FTDI-Adapter verbunden, damit ich ihn problemlos mit eigener Software bespielen konnte:



Der zugehörige Schaltplan sieht so aus:



Die Einsatzzwecke sind sehr vielseitig, viele Libraries, die ursprünglich für den Arduino entwickelt wurden, sind bereits für den ESP8266 angepasst worden, sodass die Programmierung über die Arduino-IDE die Ansteuerung externer Hardware wie Funksender oder IR-Dioden sehr einfach gestaltet.

In meinen ersten Stunden mit dem ESP8266 konnte ich daher bereits die Steuerung von Funksteckdosen über ein Webinterface samt Statusspeicherung der einzelnen Dosen (analog zur [Umsetzung mit](https://alexbloggt.com/funksteckdosen-raspberry-pi-teil1/)

dem Raspberry Pi</a>) realisieren sowie die Ansteuerung meines TV-Ger&#228;tes &#252;ber eine IR-Diode (wobei es hier zu Problemen mit der Stromversorgung trotz Kondensator kam, sodass ich die Schaltung hierzu noch umbauen werde). Auch l&#228;sst sich damit das <a href=„<https://alexbloggt.com/infrarot-geraete-vernetzen/>“ target=„\_blank“>&#196;quivalent der Umsetzung mit Arduino und Raspberry / Banana Pi</a> ersetzen. Tutorials mit vollst&#228;ndigem Programmcode und Schaltbildern zu diesen beiden Umsetzungen werden demn&#228;chst hier im Blog folgen.</p><hr/><p>Ich hoffe, ich konnte euch hiermit einen kleinen &#220;berblick &#252;ber den ESP8266 geben. Gerade f&#252;r das &#8222;Internet of things&#8220; eignet sich dieses Bauteil meiner Meinung nach sehr gut, da man damit eigene Aktoren oder Sensoren ans Netz anbinden kann &#8211; und das zu einem &#228;u&#223;erst g&#252;nstigen Preis.<br/>Bei Fragen, Anregungen etc. k&#246;nnst ihr nat&#252;rlich wie immer das Kommentarfeld unten benutzen.</p><hr/><h3>Linkliste</h3><ul><li><a href=„<http://neilkolban.com/tech/esp8266/>“ target=„\_blank“>Kolban&#8217;s book on the ESP8266 (gratis eBook zum ESP8266)</a></li><li><a href=„<https://www.mikrocontroller.net/articles/ESP8266>“ target=„\_blank“>ESP8266 (mikrocontroller.net)</a></li><li><a href=„<http://arduino-hannover.de/2014/12/11/wifi-kochbuch-mit-esp8266/>“ target=„\_blank“>WiFi Kochbuch mit ESP8266</a></li><li><a href=„<https://hackaday.com/2015/03/18/how-to-directly-program-an-inexpensive-esp8266-wifi-module/>“ target=„\_blank“>How to Directly Program an Inexpensive ESP8266 WiFi Module | Hackaday</a></li><li><a href=„<https://github.com/search?utf8=%E2%9C%93&q=esp8266>“ target=„\_blank“>Projekte zum ESP8266 auf GitHub</a></li></ul><hr/></html>

From:

<https://schnipsl.qgelm.de/> - Qgelm

Permanent link:

<https://schnipsl.qgelm.de/doku.php?id=wallabag:esp8266--erste-erfahrungen>

Last update: **2021/12/06 15:24**

