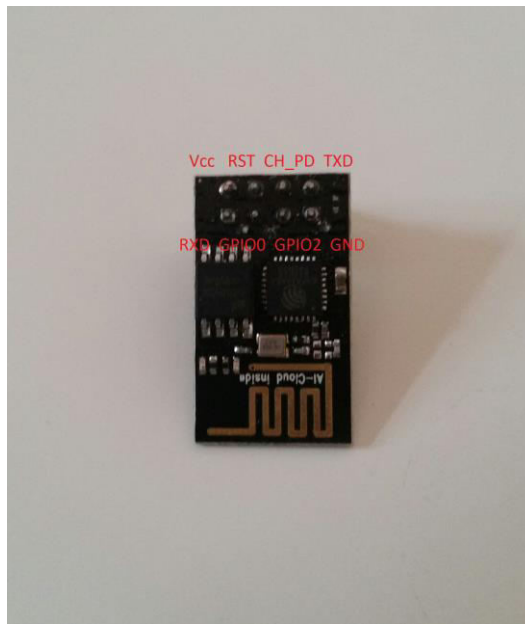


AZDelivery esp8266 01

1. Pinout und Setup des esp8266

Der AZDelivery esp8266 01 hat acht Pins, wovon zwei für die Stromversorgung (VCC und GND), zwei für die serielle Kommunikation (RXD und TXD) beispielsweise mit dem Arduino und zwei als GPIO (GPIO 0 und GPIO 2) verwendet werden. Außerdem besitzt das WiFi Modul einen Reset Pin sowie CH_PD zum Neustarten.



VCC und GND des 8266 01 müssen mit VCC und GND des Stromversorgungsmoduls beziehungsweise des Arduinos verbunden werden, wobei es sich immer empfiehlt, auch wenn man den esp8266 01 mit dem Arduino verwendet, zusätzlich eine externe Spannungsquelle zu verwenden, da das WiFi Modul bei Datenübertragungen kurzzeitig bis ca. 250mA benötigen kann, insbesondere bei Übertragungen über großen Distanzen, was den Arduino versorgungstechnisch überlasten, und Fehler hervorrufen, könnte. Des Weiteren ist zu beachten, dass das Gerät nur 3.3V unterstützt.

Im Normalbetrieb sollte man den Reset Pin über einen Pull-down Widerstand LOW schalten, und CH_PD über einen Pull-up Widerstand auf HIGH schalten. Möchte Sie den AZDelivery esp8266 01 neustarten, so müssen Sie den Reset Pin kurzzeitig auf HIGH schalten, beispielsweise über einen externen Reset Button, welcher über einen Widerstand mit +3.3V verbunden ist.

Sollte man den esp8266 mit einem Arduino oder Raspberry Pi verbinden wollen, so muss man TXD mit RX und RXD mit TX verbinden.

Über GPIO 0 und GPIO 2 kann man Sensoren oder Anderes mit dem WiFi Modul verbinden.

2. Verwendung des esp8266 01 als Wlan Modul für den Arduino

2.1 Grundlagen der Kommunikation über AT Befehle

Das AZDelivery esp8266 Wlan Modul hat standardmäßig eine Firmware vorinstalliert, mit Hilfe derer die angeschlossenen Geräte, beispielsweise ein Arduino, mit ihm über

AT Befehle kommunizieren können. Der AT Befehlssatz ist relativ alt (ca. 30 Jahre) und wurde ursprünglich für die Kommunikation mit Modems entwickelt. Durch seine Simplität und seine Sparsamkeit bezüglich Ressourcen ist er jedoch ideal für die Kommunikation mit dem esp8266 geeignet.

Alle AT Befehle beginnen mit „AT+“ und anschließend kommt erst der eigentliche Befehl. Außerdem wichtig ist, dass alle AT Befehle mit einem Zeilenumbruch (CR und NL) enden müssen. Aus diesem Grund sollte man, wenn man über die serielle Verbindung zwischen esp8266 und Arduino einen Befehl sendet immer `println` verwenden, um den Zeilenumbruch mitzuschicken, und dem Modul mitzuteilen den Befehl auszuführen.

2.2 Download einer Website

In diesem Beispiel wollen wir dem AZDelivery esp8266 über AT Befehle mitteilen eine Website herunterzuladen. Dazu muss das Wlan Modul zunächst über den Befehl `AT+CWMODE=1` in den „Wlan Client“ Modus versetzt werden. Anschließend kann es über `AT+CWJAP=“SSID“,“Passwort“` mit einem Wlan Netzwerk verbunden werden.

Nun kann man über `AT+CIPSTART=“TCP“,“website.de“,80` eine Verbindung mit dem Zielsever aufbauen. Anschließend muss man dem WiFi Modul über `AT+CIPSEND=ZAHL` mitteilen, wie lange die Anfrage ist, welche man an den Server schicken möchte. Dabei muss man immer noch ein paar Zeichen dazurechnen, da auch die Zeilenumbrüche am Ende mitgerechnet werden müssen.

Die Anfrage an den Server erfolgt nach dem Prinzip `GET /Pfad_zur_Website.html`. Nun muss man evtl. noch Leerzeichen einfügen, bis man die vorher angegebene Zahl an Zeichen erreicht hat, woraufhin der esp8266 die Anfrage automatisch an den Server schickt, und automatisch die Antwort (HTML Code der Website) zurück an den Arduino sendet. Solange die angegebene Zahl an Zeichen nicht erreicht ist, tut der esp8266 nichts. Sollte die Anfrage länger sein, als angegeben gibt er „busy“ aus.

Aufgrund der Komplexität dieses Verfahrens sollte man es zunächst am PC über die serielle Konsole ausprobieren, bevor man das Verfahren in Arduino Code übersetzt, da sich ansonsten die Fehlersuche sehr schwer gestaltet.

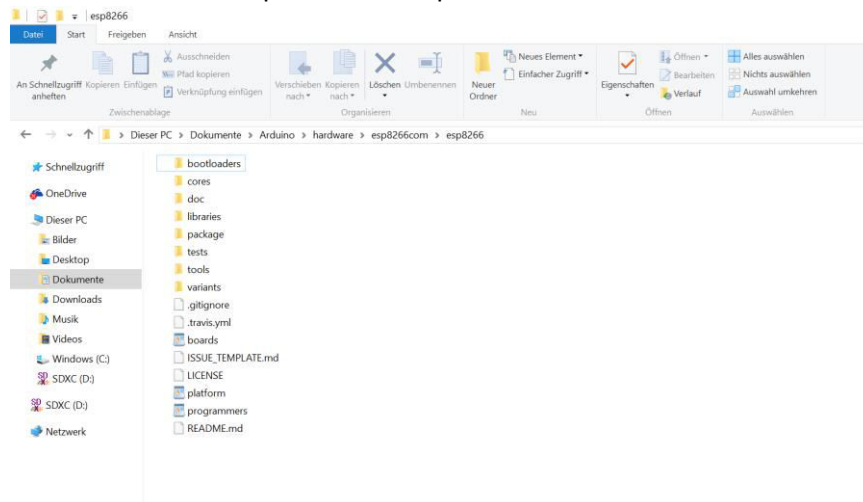
3. Autonomer Betrieb des esp8266 01

3.1 Vorbereitung der Arduino IDE

Zusätzlich zur Verwendung als einfaches Wlan Modul für den Arduino, ist es auch möglich den AZDelivery esp8266 01 autonom zu betreiben, also auch das Kernprogramm auf ihm

ausführen. Dabei kann man ihn praktischerweise direkt in der Arduino IDE mit all ihren Gepflogenheiten programmieren.

Dazu muss man jedoch einen Ordner auf Github (<https://github.com/esp8266/Arduino>) herunterladen, welcher benötigt wird, damit die Arduino IDE den esp8266 erkennt und mit ihm arbeiten kann. Den Inhalt des .zip Ordners muss man nun in den Arduino Ordner im Unterordner hardware unter esp8266com abspeichern.



3.2 Programmierung des esp8266 01

3.2.1 Download einer Website

In diesem Beispiel geht es darum den AZDelivery esp8266 01 so zu programmieren, dass er eine Website herunterlädt. Dadurch kann man anschließend eine beliebige Information aus der heruntergeladenen Website herausholen.

Um eine Website herunterzuladen, muss das WiFi Modul sich zunächst mit einem bestehenden WiFi Netzwerk verbinden, welches über eine Internetverbindung verfügt. Anschließend erzeugt man ein Objekt vom Typ WiFiClient, welches die Kommunikation mit dem Server übernimmt, und an ihn die Anfrage sendet, ihm die Website zu schicken.

Nun speichert der WiFiClient die empfangenen Daten in einen String, welcher den gesamten HTML Code der Website, nicht nur den angezeigten Text, den man sieht, wenn man die Website im Browser ansieht, in einem String. Diesen String kann man nun über die einschlägigen Befehle nach Belieben bearbeiten und dursuchen.

Auf diesem Prinzip basierend gibt es bereits viele Beispielsketches, unter anderem unter Datei > Beispiele > ESP8266WiFi > WiFiClient.

3.2.2 esp8266 01 als Webserver

Der AZDelivery esp8266 01 kann auch als regulärer Webserver verwendet werden. Dabei empfiehlt es sich jedoch nicht ihn für eine reguläre Internetseite zu verwenden, da seine Hardwareressourcen zugunsten anderer Faktoren, wie beispielsweise geringem Energieverbrauch oder einem niedrigen Preis, gering sind. Er ist jedoch ideal geeignet, um Informationen, beispielsweise von angeschlossenen Sensoren, bereitzustellen, sowie Nutzereingaben für angeschlossene Geräte (z.B. LED, Motor, ...) anzunehmen.

Um das Wlan Modul als Webserver zu verwenden, muss man es mit einem bestehenden WiFi Netzwerk verbinden, oder ein Neues erstellen. Anschließend muss man ein Objekt vom Typ

WiFiServer erstellen, welches auf eingehende Anfragen auf dem angegebenen Port (typischerweise 80) wartet, und an den Client den gespeicherten (HTML)-Code/Text schickt.

Einen einfachen Beispielsketch dafür gibt es unter Datei > Beispiele > ESP8266WiFi > WiFiWebServer.