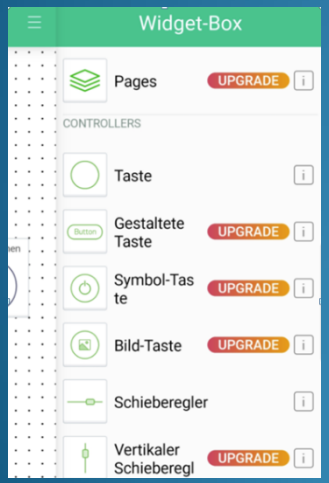
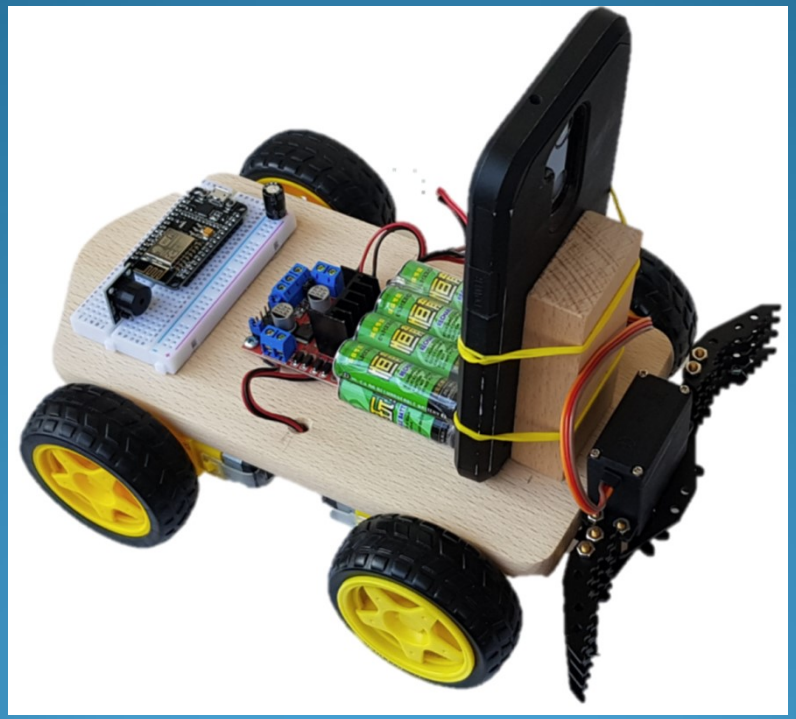
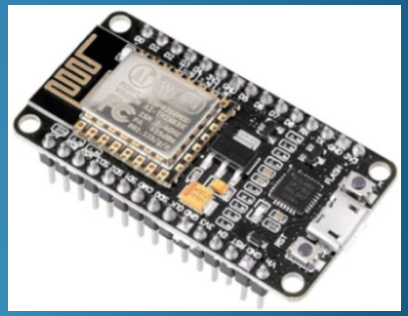
GTA Mechatronik Teil 2

Wir steuern das Roboterauto über das Internet
und lernen dabei viel über HTML, IoT und das WWW





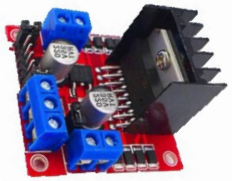


Inhaltsverzeichnis

Roboterauto Komponenten

ESP8266 NodeMCU (V2)
mit WLAN-Antenne

L298N DC Motor
Driver Module

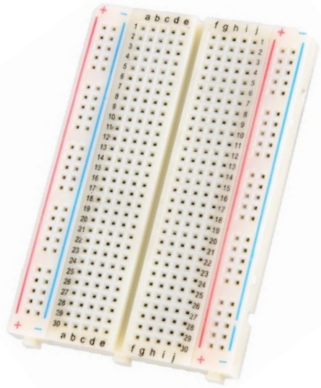


NiMH Akku-Pack 9,6V
800 oder 2400 mAh

Tamiya Female

Smartphone

Breadboard

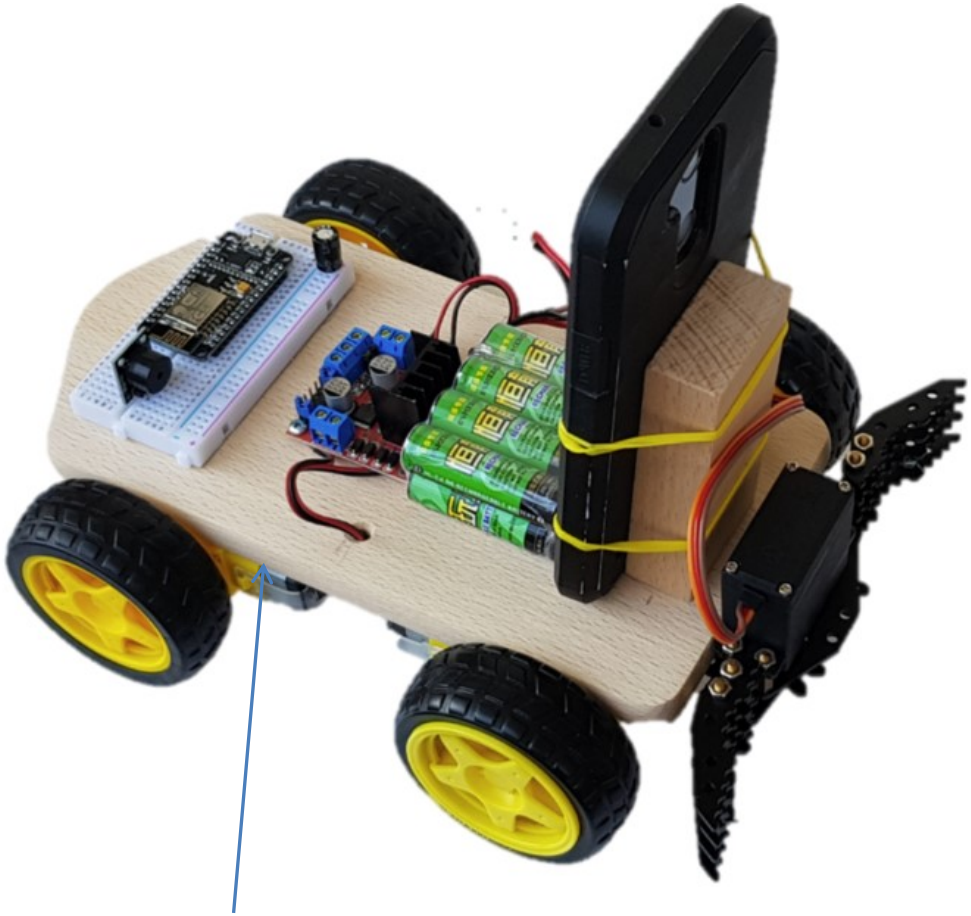


Servo MG996R
mit Greifer

Buzzer
(Piezo-Summer)

Radantriebe
4x

Roboterauto montiert (ohne Verdrahtung)



Heißkleber (vorher Kunststoffoberfläche mit Sandpapier anrauhen)

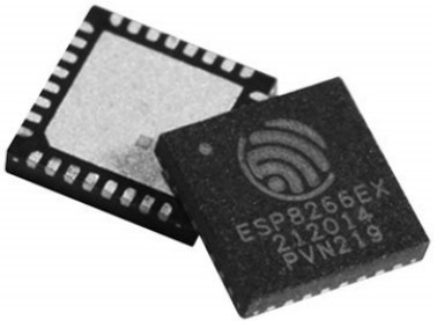
Espressif ESP8266

Espressif Systems aus China bietet einen Chip an, der bereits WLAN-Funktionaltät integriert hat (für 2,4 GHz).

Die Bezeichnung des Chip ist **ESP8266** bzw. die Weiterentwicklung **ESP32** (noch leistungsfähiger, aber höherer Stromverbrauch, siehe Anlage).

Der Chip ist in einem in einem 32pin-QFN Gehäuse untergebracht.

https://www.espressif.com/en



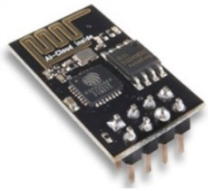
About Espressif

Espressif Systems is a multinational, fabless semiconductor company established in 2008, with headquarters in Shanghai and offices in Greater China, India and Europe. We have a passionate team of engineers and scientists from all over the world, focused on developing cutting-edge WiFi-and-Bluetooth, low-power IoT solutions. We have created the popular ESP8266 and ESP32 series of chips, modules and development boards. By leveraging wireless computing, we provide green, versatile and cost-effective chipsets.

ESP8266 -> ESP-12E -> NodeMCU V2 Board

Es gibt mehrere Module mit dem Chip ESP8266.

Zum Beispiel: ESP-01 oder ESP-12E



ESP-01

ESP-12E

Die etwas größeren Module (z.B. ESP-12E) haben auch zahlreiche I/O-Ports und serielle Schnittstellen wie SPI, I^2C und eine asynchrone serielle Schnittstelle UART.

Auf Basis des Moduls ESP-12E gibt es auch ein komplettes Entwicklungs-Board.

Es hat die Bezeichnung „NodeMCU V1.0“ (in der Arduino IDE) bzw. „NodeMCU V2“.

Das NodeMCU hat eine USB-Schnittstelle und eine Spannungs-Stabilisierung 5V (Chip ESP8266 bzw. Modul ESP-12E arbeiten nur mit 3,3Vdc)

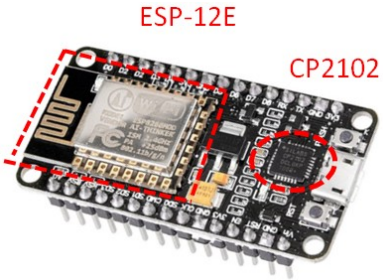
An den Ausgängen des NodeMCU liegen nur 3,3V (Arduino: 5V).

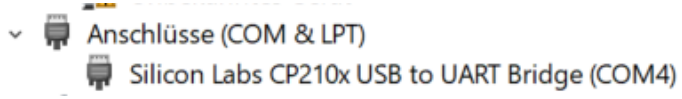
Das Pin-Raster des NodeMCU V2 ist Breadboard-kompatibel (siehe auch NodeMCU V2 und V3).

Als USB-Wandler wird der Chip CP2102 verwendet.

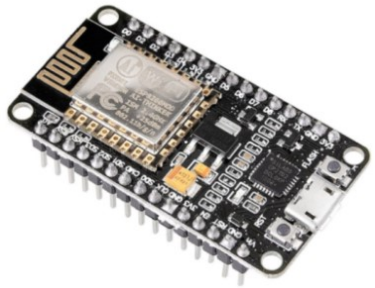
Wenn der dafür Treiber noch nicht installiert ist – siehe ESP8266 – Treiber für CP2102 .

Ein empfohlener 100 µF Kondensator zwischen VCC und GND ist beim NodeMCU Board bereits vorhanden.



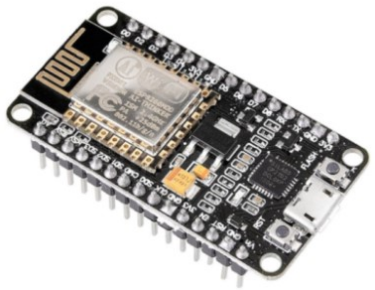


ESP8266 (NodeMCU) versus Arduino Nano



WLAN fähig	nein	ja , für 2,4 GHz
CPU	Atmel ATmega 328P	Tensilica Xtensa LX106
	8 bit	32 Bit
Flash-Speicher	32 kByte	4 Mbyte
CPU-Takt	16 MHz	80 MHz
A/D Wandler	8 Bit	10 Bit
Preis	11 €	5 €
	(2021 deutlich teurer geworden)	
Hochladezeit der Sketche	kurz	lang, deshalb Upload Speed höher einstellen:
Reset (Neustart) bei Aufruf des Serial Monitor ?	Ja	Nein

ESP8266 (NodeMCU) versus Arduino Nano



Spannungsversorgung	5V an „5V“	3,3V an „3.3V“
	7…12V an „VIN“	5V...9V an „VIN“
	USB-Kabel Mini	USB-Kabel Micro
Stromaufnahme	niedriger	höher , vor allem bei WLAN-Betrieb im „Deep-Sleep-Modus“ absenkbar 25 µA bis 400 mA, durchschnittlich 80 mA
Strombelastbarkeit per Pin	20...40 mA	12 mA source current ; 20 mA sink current
Strombelastbarkeit insgesamt	200 mA	an jedem GPIO siehe oben
I/O Pins nur digital (GPIO)	14 (0 / 5V)	13 (0 / 3,3V) Eingänge Motortreiber L298N: High min 2,3V
davon geeignet für PWM	6	13
I/O Pins nur analog (GPIO)	2 (0...5V)	1 (0...3,3V an Pin A0)
I/O Pins digital/analog (GPIO)	6	nein

https://www.youtube.com/results?search_query=esp8266+vs+arduino+nano

NodeMCU Lua Amica Modul V2 ESP8266 ESP-12E mit CP2102

Pinbelegung NodeMCU -Board (V2)

Mikrocontroller - Elektronik.de

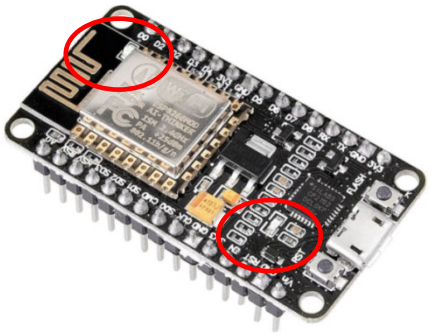
TOUT	ADC0	A0	D0	GPIO16	- USER - WAKE
	Reserve		D1	GPIO5	
	Reserve		D2	GPIO4	
SDD3	GPIO10	D12	D3	GPIO0	- FLASH
SDD2	GPIO9	D11	D4	GPIO2	- TXD1
	SDD1	INT	3,3V		
	SDCMD	MOSI	GND		
	SDD0	MISO	D5	GPIO14	- HSPICLK
	SDCLK	SCLK	D6	GPIO12	- HSPIQ
	GND		D7	GPIO13	- RXD2 - HSPID
	3,3V		D8	GPIO15	- TXD2 - HSPID
	EN		D9	GPIO3	- RXD0
	RESET		D10	GPIO1	- TXD0
	GND		GND		
	POWER 5,0V		3,3V		

USB

Programmierung und Spannungsversorgung

Diese LED (blau) ist mit D4 verbunden
- blinkt schnell während des Upload
- blinkt einmalig bei Reset-Taster loslassen
- kann ein/ausgeschaltet werden mit

```
pinMode (D4,OUTPUT);
digitalWrite (D4,LOW);      // Ein
digitalWrite (D4,HIGH);     // Aus
```



Diese LED (blau) ist mit D0 verbunden

Für die Programmierung kann man entweder die NodeMCU – Nr. oder den GPIO-Port angeben.
Wenn nur Zahlen im Sketch angegeben werden, dann ist das immer die GPIO-Nummer.
Sonst muß das D davor . Zum Beispiel D1 entspricht dem Port GPIO 05.

https://www.az-delivery.de/collections/wifi-module/products/nodemcu?ls=de

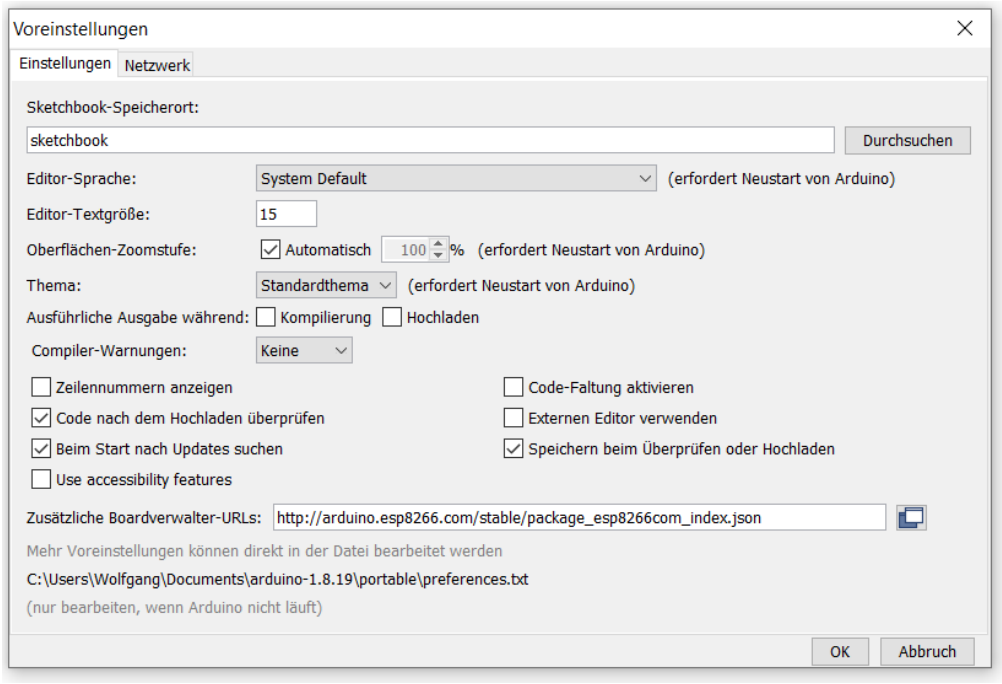
ESP8266 programmieren mit der Arduino IDE

Wenn die Arduino-IDE installiert ist und ein Ordner „sketchbook“ angelegt ist (siehe Teil 1):

Öffnen der Arduino IDE.
Im Feld „Zusätzliche Bordverwalter-URLs“ eintragen:

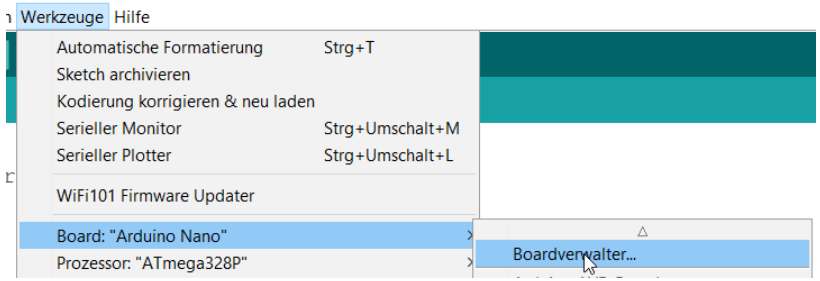
https://arduino.esp8266.com/stable/package_esp8266com_index.json

und dann OK.



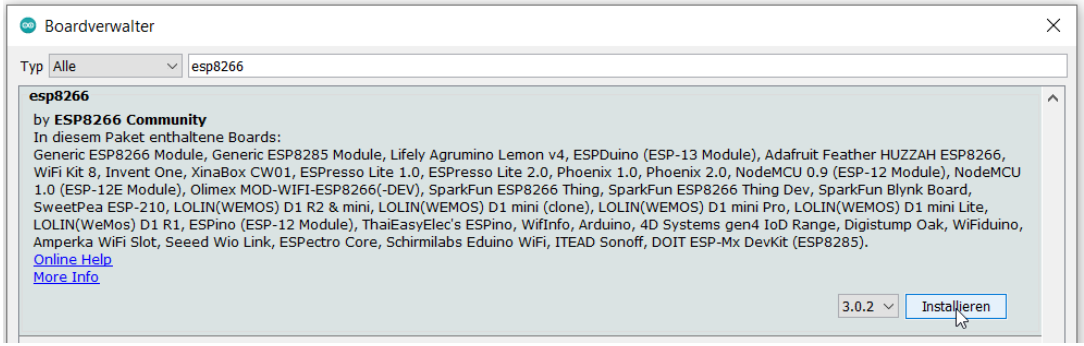
ESP8266 programmieren mit der Arduino IDE

Um den NodeMCU ESP8266 hinzuzufügen, klicke unter „Werkzeuge“ -> „Board“ auf „Bordverwalter“

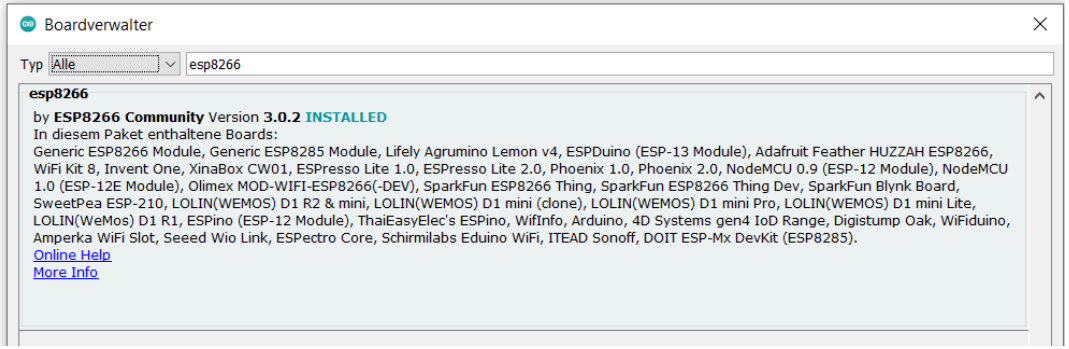


und gib ein „ESP8266“.

Klicke auf installieren (es muß eine Internetverbindung bestehen).

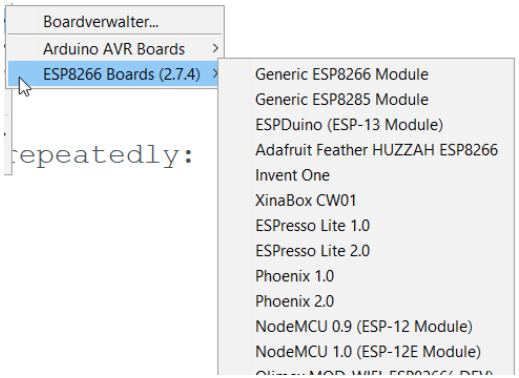


Die Installation dauert einige Minuten, der Verlauf wird angezeigt.

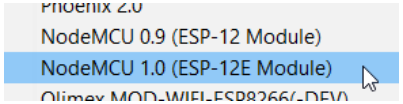


ESP8266 programmieren mit der Arduino IDE

Im Boardverwalter sieht man jetzt,
daß eine Reihe neuer Boards installiert wurde.

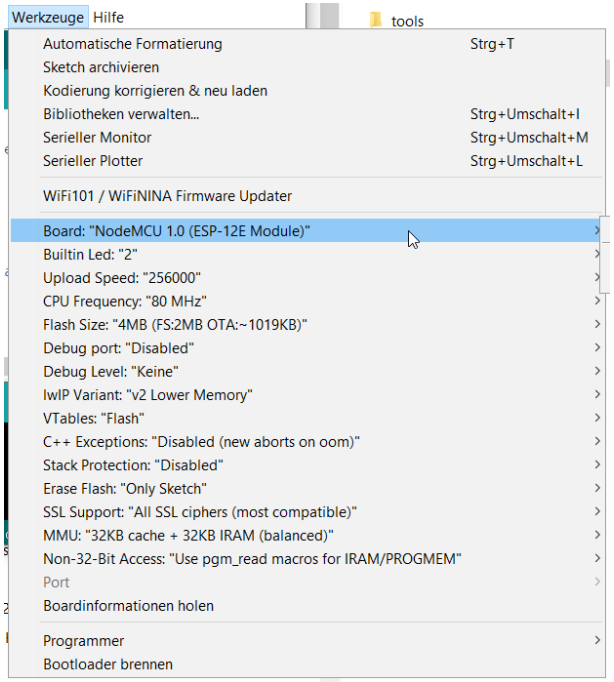


Wir klicken an:



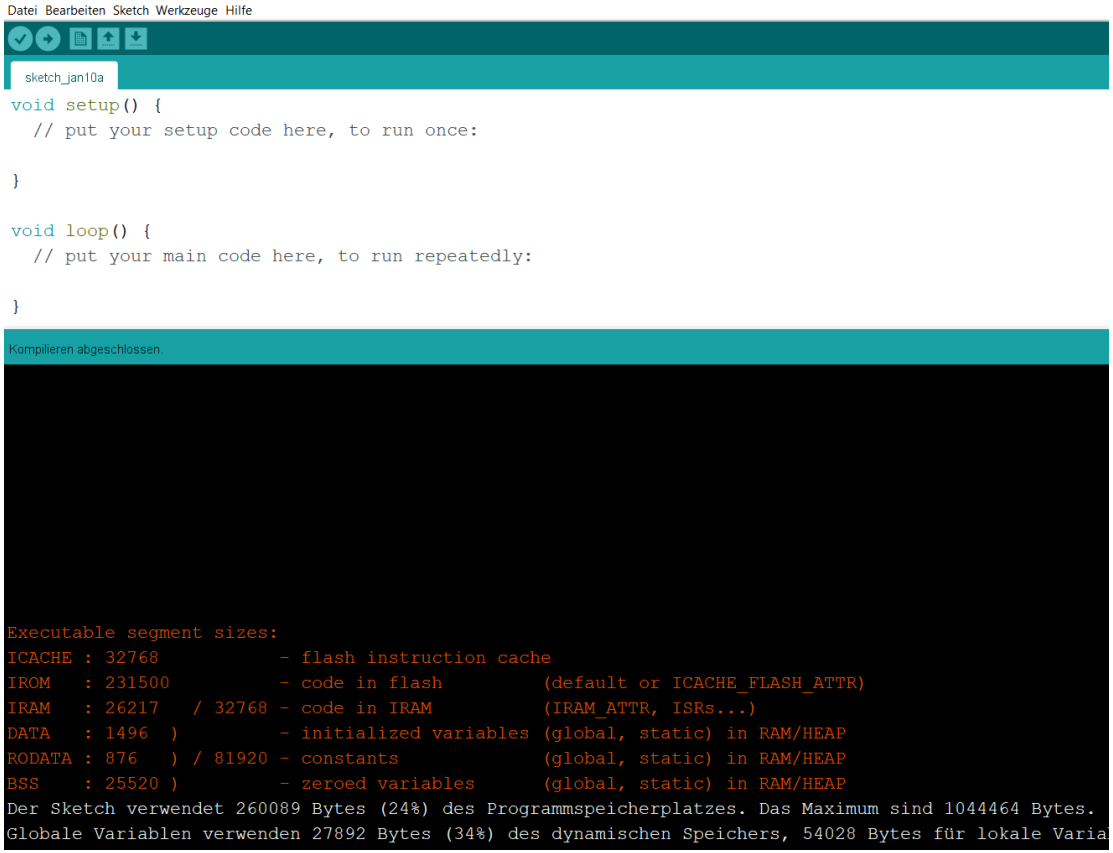
Jetzt erscheinen einige Einstellungen
(die teilweise auch geändert werden könnten):

Wichtig, wegen langer Upload-Zeit beim ESP8266:
Einstellung einer höheren Upload Speed. z.B.: 256000



ESP8266 programmieren mit der Arduino IDE

Nach Start Kompilieren  sollte erscheinen:



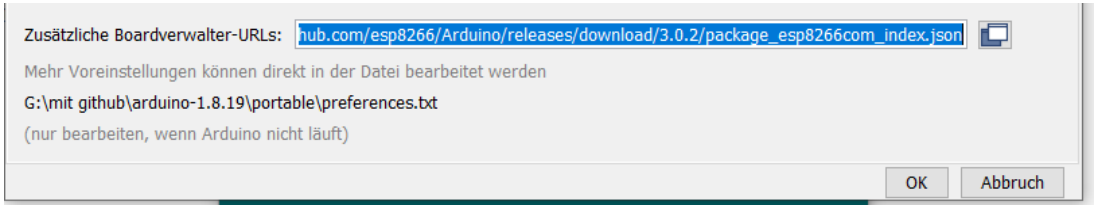
Das Kompilieren dauert wesentlich länger als beim Arduino.

ESP8266 programmieren mit der Arduino IDE

Falls die folgende Fehlermeldung erscheint:

-> Start beim ersten Kompilieren mit Internetverbindung

Oder die Boardverwalter-URL von der Website von github eintragen (evtl eine frühere Version, z.B. 2.7.4):



https://github.com/esp8266/Arduino/releases/download/3.0.2/package_esp8266com_index.json

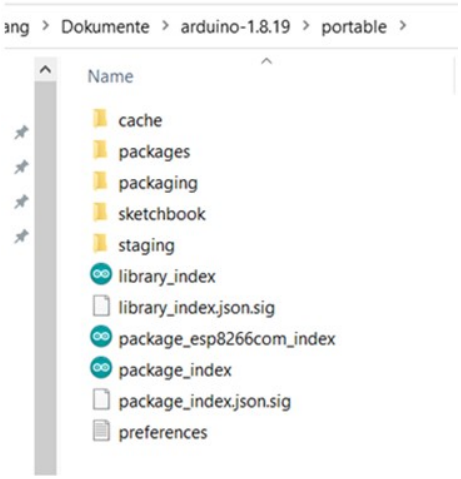
Siehe dazu:
https://www.youtube.com/watch?v=_PLfl_cgIHs

https://github.com/esp8266/Arduino/releases/

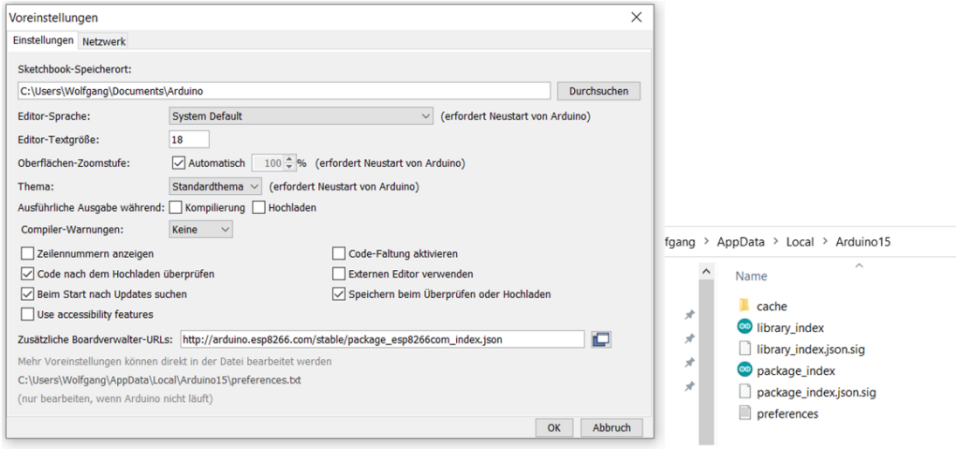
ESP8266 programmieren mit der Arduino IDE

Übrigens, bei anklicken in „Voreinstellungen:

Sieht man die jetzt angelegten Ordner und Dateien.
Sie befinden sich im Ordner „portable“.

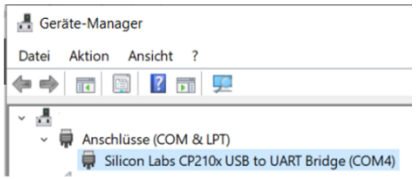


Wenn kein Ordner „portable“ angelegt worden ist und die Arduino IDE installiert wurde auf C:\Program Files (x86) :
Die jetzt angelegten Dateien und Ordner befinden sich auf „Benutzer/AppData/Local/Arduino15:



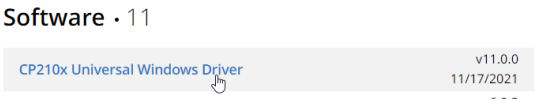
ESP8266 – Treiber für CP2102

Prüfen, ob der Treiber für den USB/UART–Wandler CP2102 bereits installiert ist (ESP8266 muß angeschlossen sein):

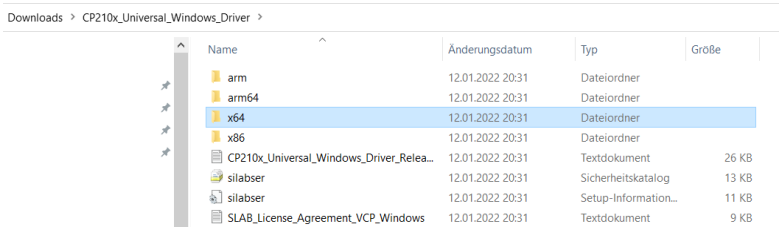


Wenn nicht, download von Silicon Labs: (als zip-Datei)

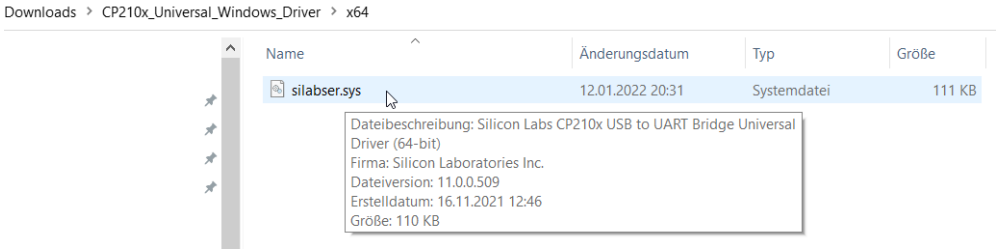
https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers



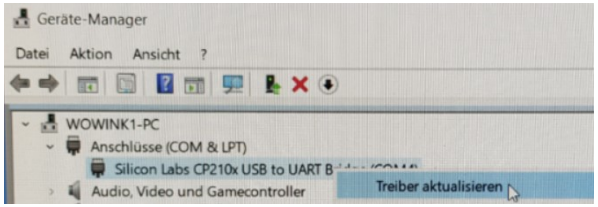
Dann Extrahieren.



Die Datei silabser.sys in den Ordner C:\Windows\System32\drivers laden.



Oder im Geräte-Manager downloaden bzw. aktualisieren.



WLAN (Wi-Fi)

Wireless Local Area Network (WLAN; deutsch drahtloses lokales Netzwerk) bezeichnet ein lokales Funknetz.

In vielen Ländern ist der Begriff „WLAN“ nicht bekannt. Hier gilt der Begriff **Wi-Fi** (Wireless Fidelity) , obwohl WLAN aus dem Englischen kommt.

Öffentliche und kommerzielle WLAN-Access-Points mit Internetanbindung nennt man „**Hot Spots**“.

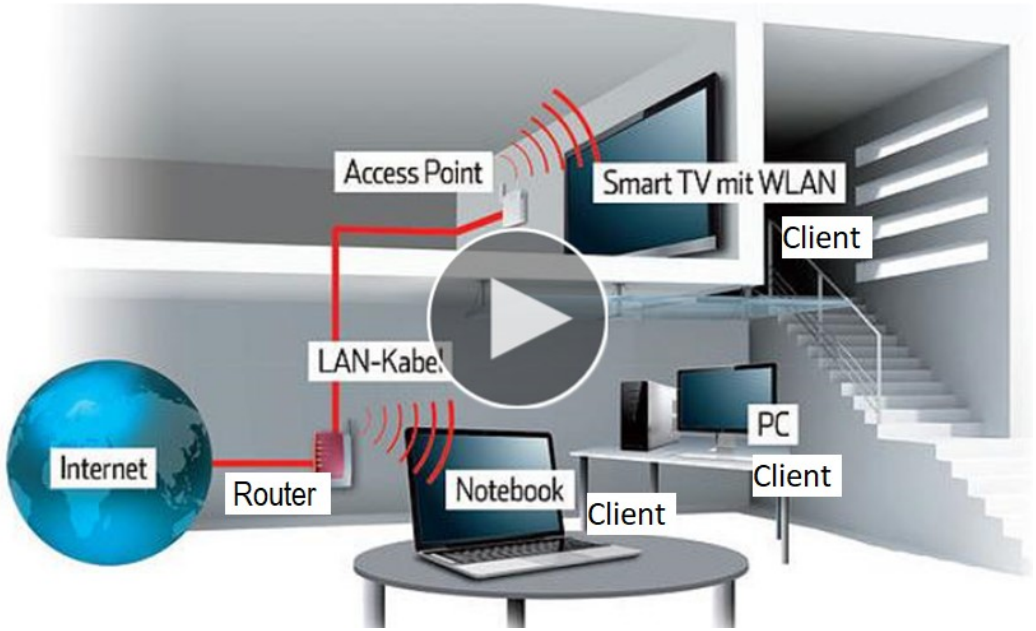
Die von WLAN-Geräten benutzten Funkfrequenzen liegen um 2,4 GHz beziehungsweise 5,4 GHz.

Die zulässige äquivalente isotrope Strahlungsleistung (EIRP) von 100 mW (2,4 GHz) beziehungsweise 500 mW (5,4 GHz) handelsüblicher 802.11-Endgeräte lässt 30 bis 100 Meter Reichweite auf freier Fläche erwarten.

Nach mehreren Studien, u. a. des BfS (Bundesamt für Strahlenschutz), gibt es innerhalb der gesetzlichen Expositionsgrenzwerte (siehe oben) nach dem aktuellen Stand der Wissenschaft keine Hinweise, dass diese hochfrequenten elektromagnetischen Felder gesundheitliche Risiken darstellen.

https://www.youtube.com/watch?v=mWKpty-YuWw
https://www.youtube.com/watch?v=sGDvulsFfYs

WLAN (Wi-Fi)



Wireless Access Point

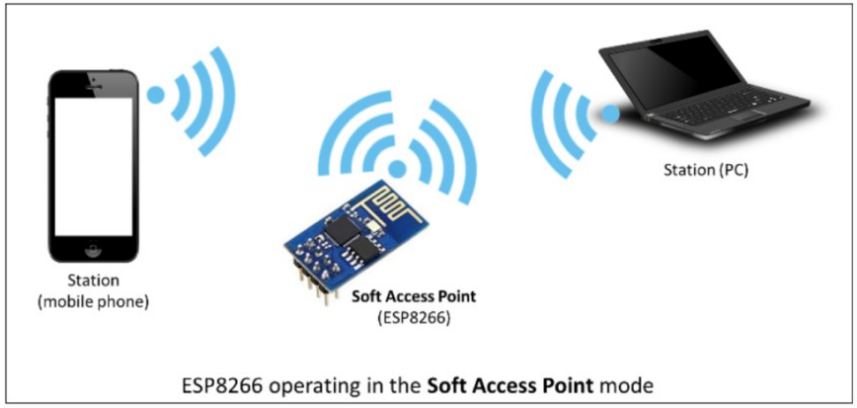
Ein ,,Wireless Access Point" oft abgekürzt zu ,,Access Point" (AP) stellt ein lokales drahtloses Netzwerk her. Geräte können sich mit dem AP verbinden.

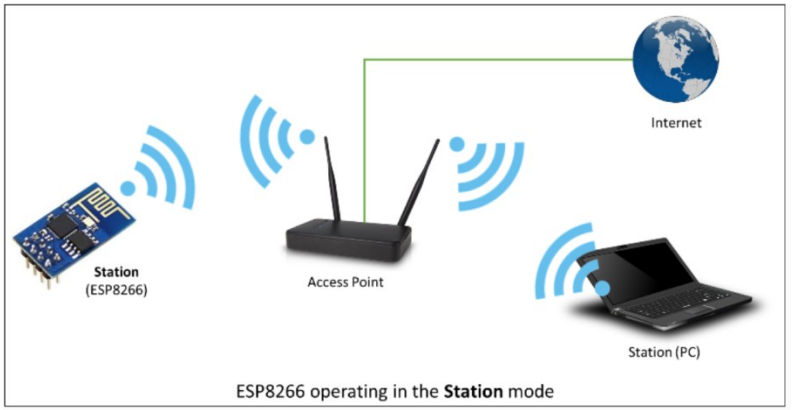
Der AP ist wiederum meist mit einem Router verbunden, der die Verbindung zum Internet herstellt.
Ein AP kann aber auch ausschliesslich ein lokales Netzwerk aufbauen, ohne Verbindung zu einem Router.

Das wird als „Soft-AP“ bezeichnet.

Station (STA) and Access Point (AP)

Der ESP8266 kann in zwei Modes arbeiten: Access Point (AP) und Station (STA) .





Als Station (Client) kann sich der ESP8266 mit einem vorhandenen WiFi-Netzwerk verbinden, das z.B. durch einen Router (Fritz-Box) zur Verfügung gestellt wird. Praktisch genauso, wie das ein Smartphone oder Computer macht.

Im AP Mode erstellt er aber auch ein eigenes WLAN Netz.

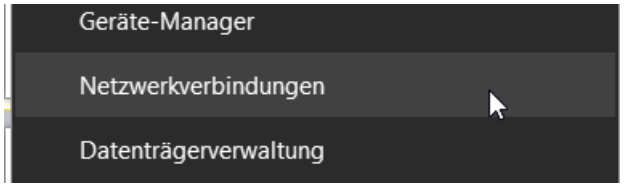
Der Unterschied zu einem Router ist nur, daß keine Internet-Verbindung besteht und auch kein LAN-Kabel angeschlossen werden kann.

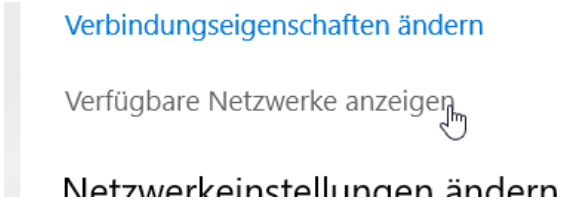
Nach Upload des Sketches bzw. Reset sind beide Modes aktiv.
Der ESP8266 stellt also automatisch auch einen Access Point zur Verfügung, den man aber in der Regel gar nicht benötigt.
Um dieses WLAN Netz (dessen SSID „ESP“ gefolgt von einer Nummer ist) zu deaktivieren, reicht eine Zeile in der Setup-Routine:

```
WiFi.mode(WIFI_STA);                    // konfiguriere ESP als  Station, also nicht als Access-Point
WiFi.hostname(HOSTNAME);
WiFi.begin(ssid, password);
```

Behandlung WLAN am Computer

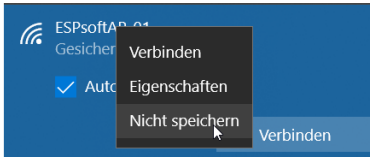
Rechte Maustaste (MT):



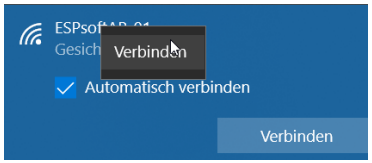


WLAN-Acess-Points werden auch noch angezeigt, wenn sie nicht mehr aktiv sind.
Um das zu prüfen, immer mit rechter MT klicken:

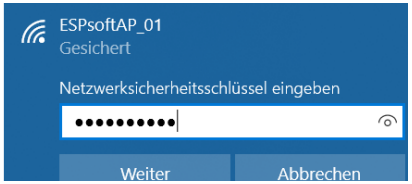
Wenn Anzeige wie rechts → „Nicht speichern“



Verbinden , wenn Anzeige

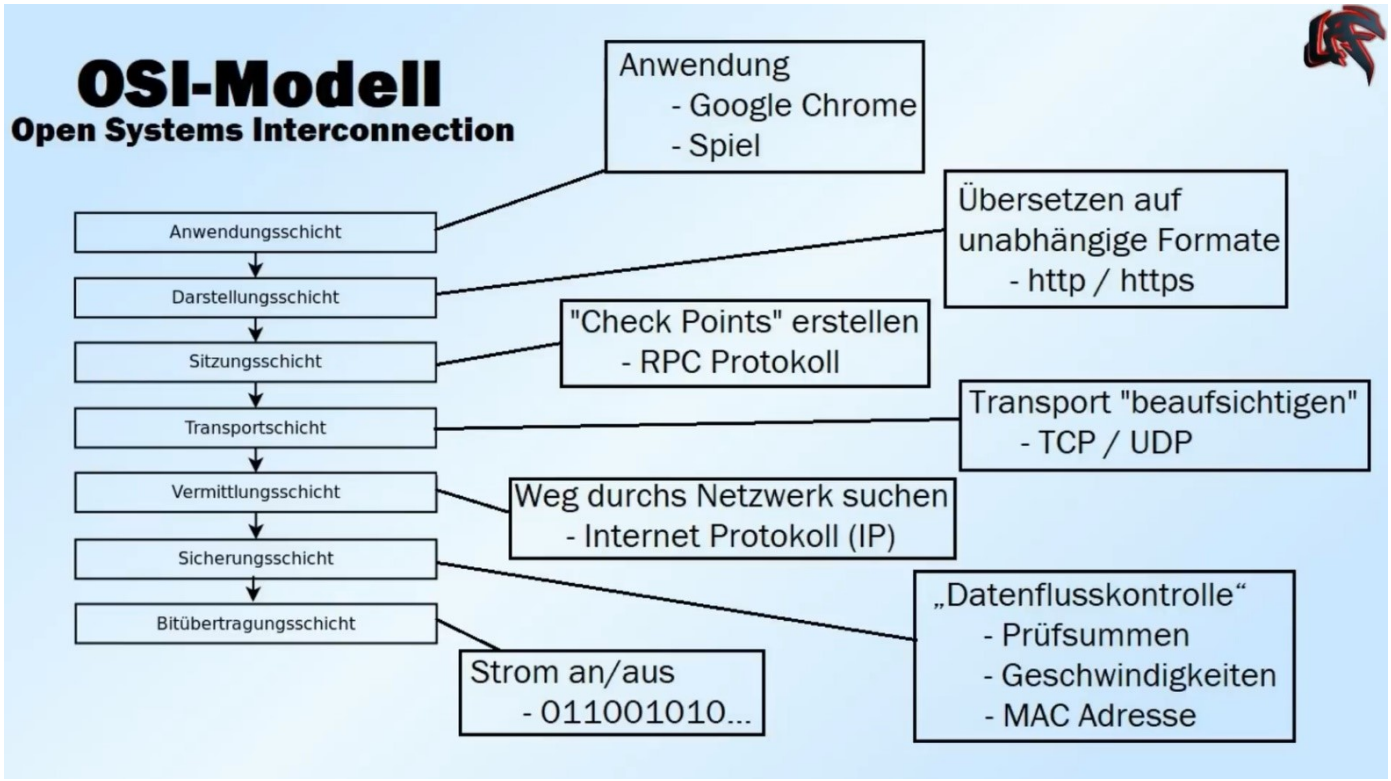


Dann muß immer neu eingegeben werden:



Kommunikation im Internet - Datenübertragung in Netzwerken

https://www.youtube.com/watch?v=xiTr5B19Zd4



https://www.youtube.com/watch?v=emmD9MhAqiM

Was ist die MAC-Adresse ?

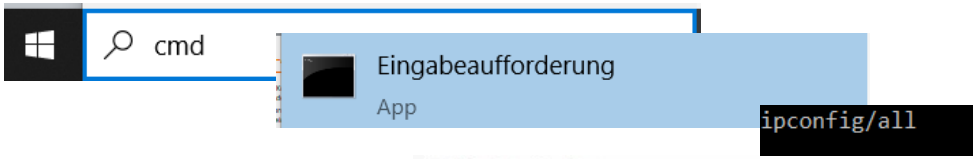
https://www.youtube.com/watch?v=gOQhZbYwLVw

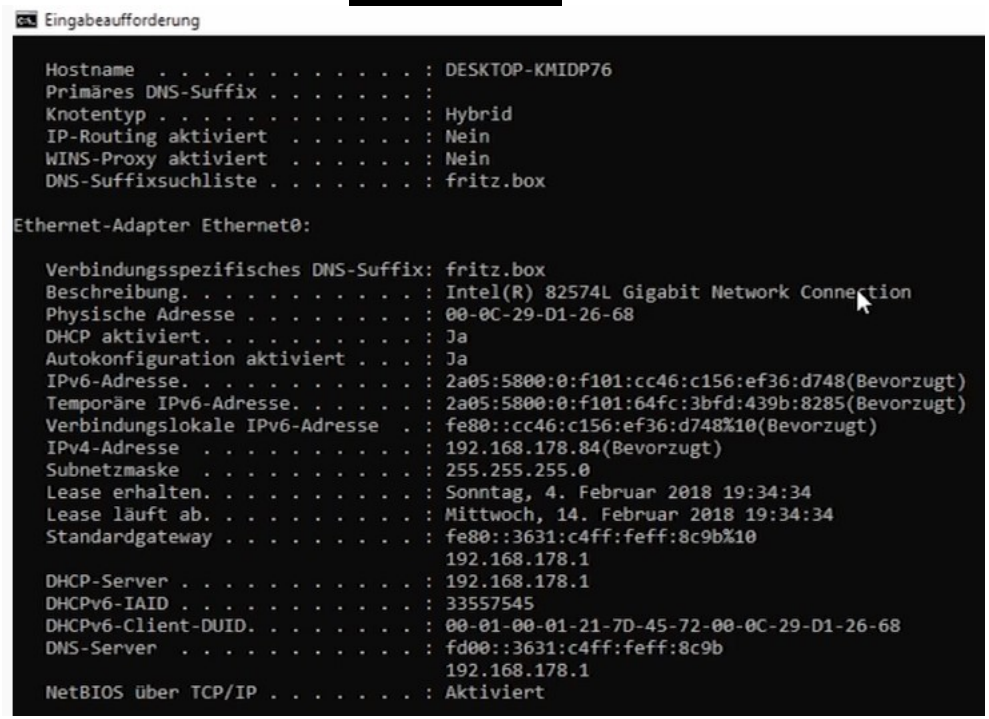
IP-Adressen im Internet und im Heimnetz

https://www.youtube.com/watch?v=5KdEc4Dto9Y

https://www.youtube.com/watch?v=BWnBOpZgXsc

Eigene IP-Adresse ermitteln (aus Video oben):





Private IP Adressen (IP-Adressen im Heimnetz)

Die IP-Adressen der Geräte, die zu Hause mit dem Router verbunden sind, bleiben dem Internet verschlossen. Diese IP-Adressen können also innerhalb eines Heimnetzes („privaten“ Netzes) von jedem Computer verwendet werden.
Eine typische Adresse ist zum Beispiel:

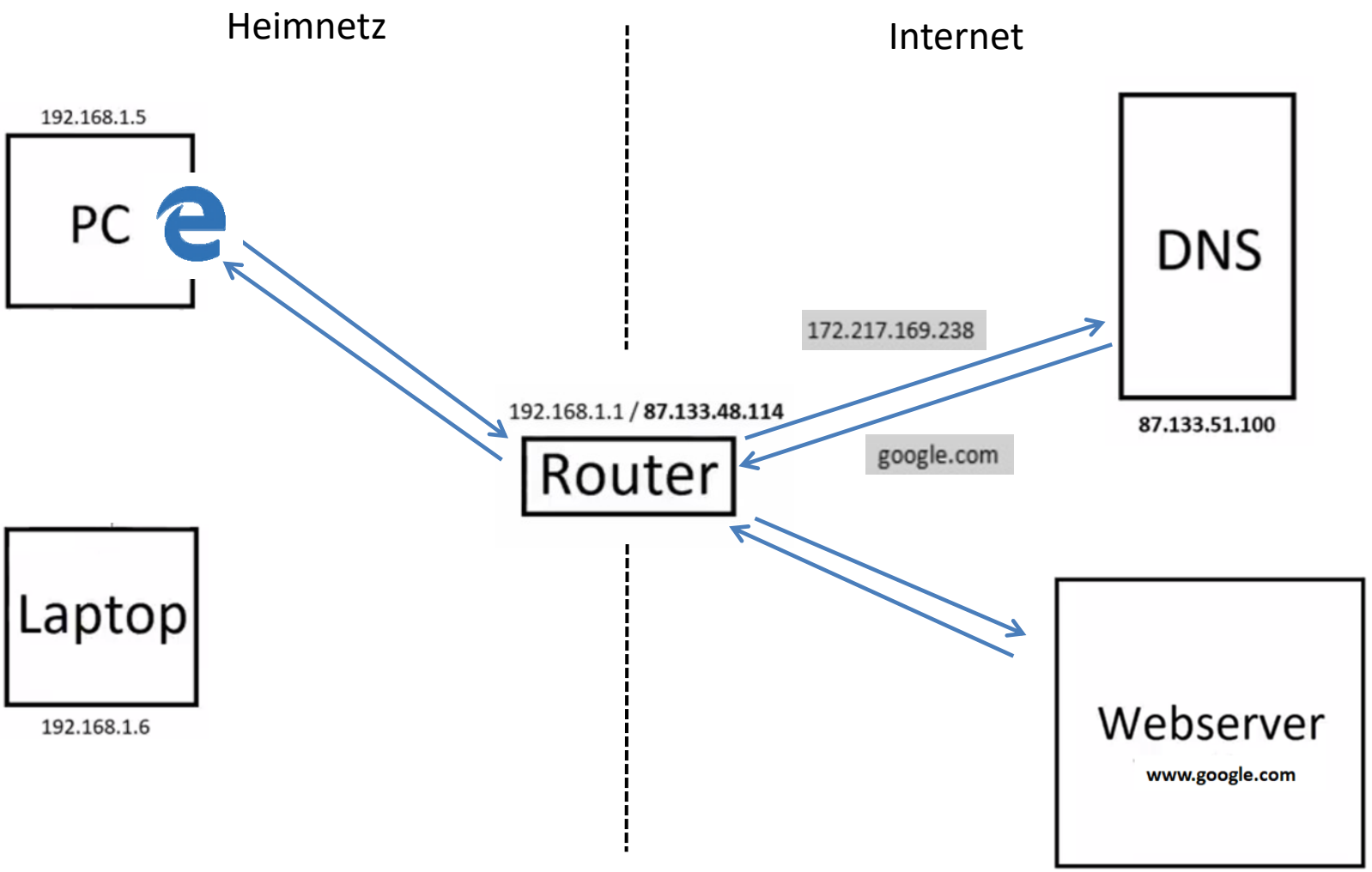
```
192.168.0.1.
```

Von der internationalen Organisation IANA wurden drei private IP-Adressbereiche festgelegt:

Die **Internet Assigned Numbers Authority** ist eine Abteilung der ICANN und für die Zuordnung von Nummern und Namen im Internet, insbesondere von IP-Adressen, zuständig. Sie ist eine der ältesten Institutionen im Internet.

Netzadressbereich
10.0.0.0 bis 10.255.255.255
172.16.0.0 bis 172.31.255.255
192.168.0.0 bis 192.168.255.255

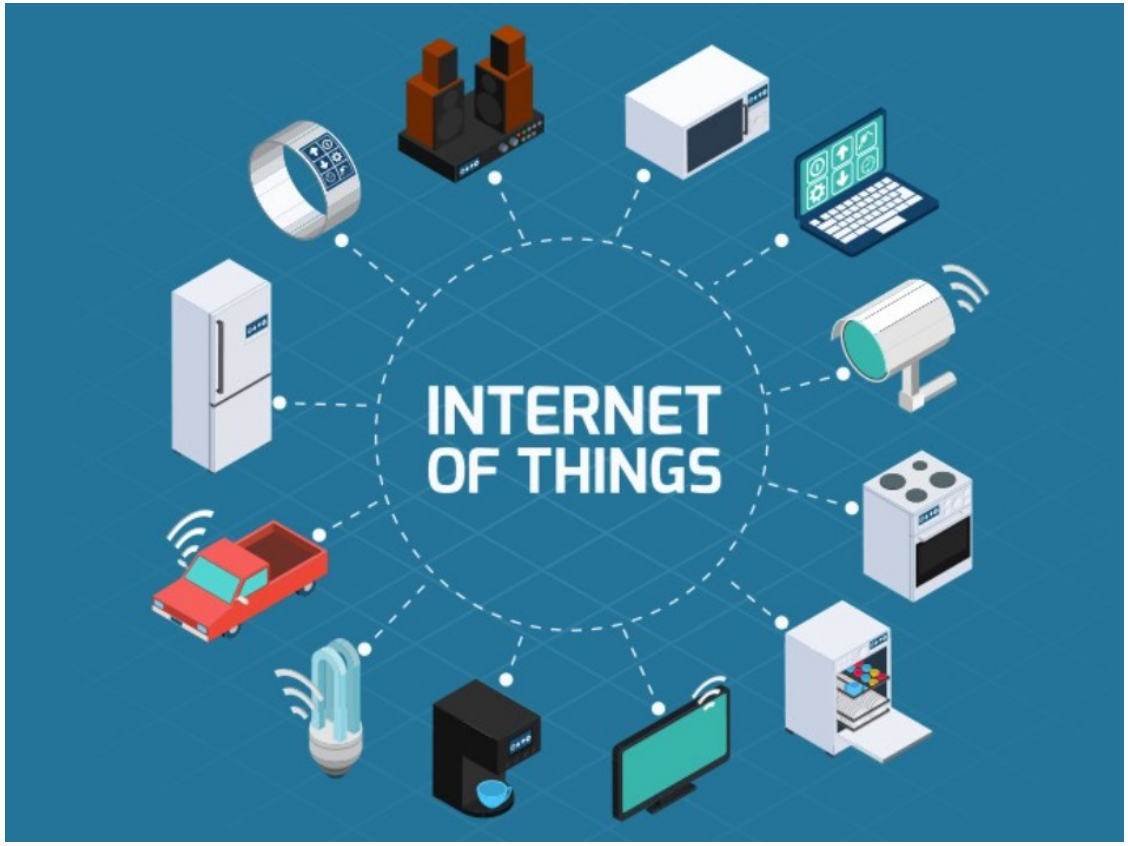
Kommunikation im Internet



IoT Internet of Things

https://www.youtube.com/watch?v=16z0o1ccDM8

https://www.youtube.com/watch?v=wDBkIToA-MQ



https://www.youtube.com/watch?v=LlhmzVL5bm8

Apps für IoT

Blynk

Thingspeak

Virtuino

Arduino IoT

- Spezielle Server in der „Cloud“ für das IoT, über die der Datentransfer abläuft
- Zur Benutzung muß man sich dort anmelden
- Dann Download der jeweiligen Software (App)
- Die Apps stellen graphische Bedienoberflächen (graphic interfaces) zur Verfügung, über die Steuersignale versendet oder Sensordaten abgerufen werden können

Programmierung von Webseiten mit HTML

Siehe https://gtamechatronik.de/Website_programmieren/

Sketche

Inhaltsverzeichnis

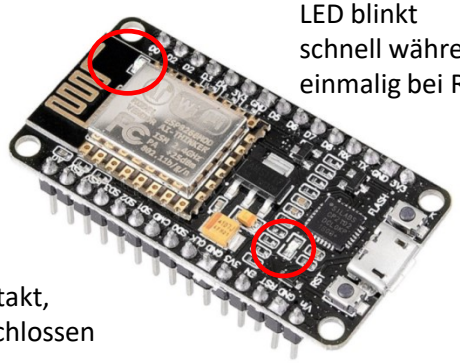
Zuordnung NodeMCU (ESP8266) Pins

ESP	Funktion	Eing/Ausg		
D0~ (GPIO16)	LED	Ausgang		Schaltet auch interne LED Für DeepSleep mit RST verbinden
D1~	Motoren rechts Brücke	Ausgang	L298N INA3	
D2~	Motoren rechts Brücke	Ausgang	L298N INA4	
D5~	Motoren rechts PWM	Ausgang	L298N ENB	PWM (analogWrite)
D3~	Motoren links Brücke	Ausgang	L298N INA1	
D4~	Motoren links Brücke	Ausgang	L298N INA2	
D6~	Motoren links PWM	Ausgang	L298N ENA	PWM (analogWrite)
D7~	Buzzer	Ausgang		
D8~	Servo Greifer	Ausgang		

~ Ausgang PWM möglich (bei allen GPIO möglich) 10bit => 0 ... 1024

Sketch 40 LED blinken

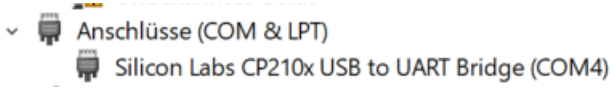
GPIO16 (an Pin D0) ist gleichzeitig auch an die Board-interne LED angeschlossen.
Der Sketch lässt diese LED im Takt von 1 s blinken.

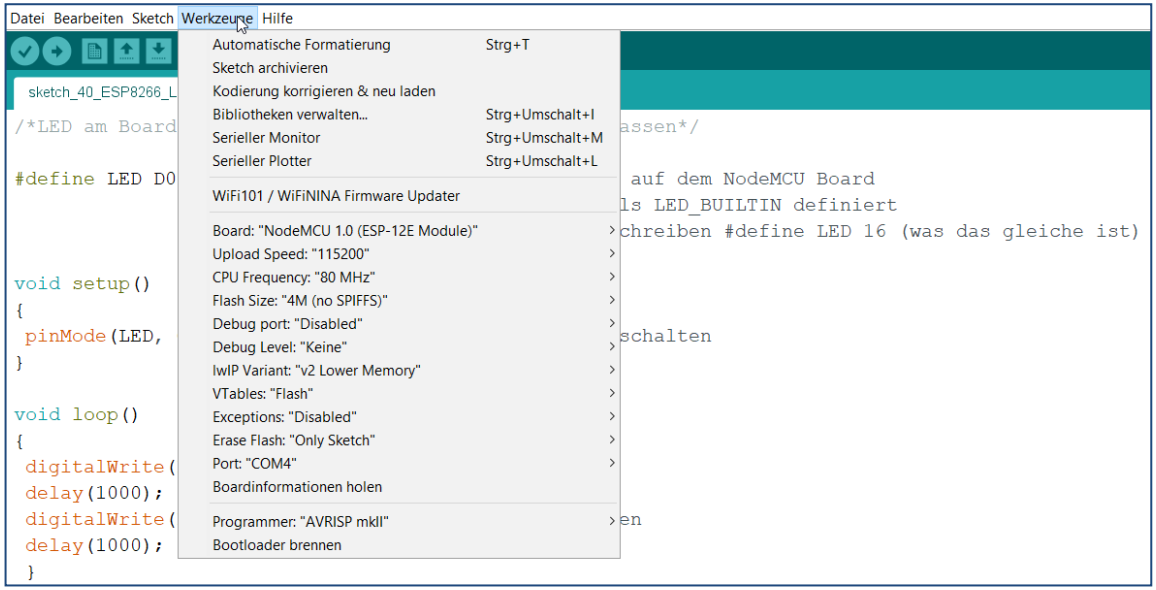


LED blinkt
schnell während des Programm-Hochladens
einmalig bei Reset-Taster loslassen

LED blinkt im Sekundentakt,
wenn Hochladen abgeschlossen

USB-Wandler Chip CP2102

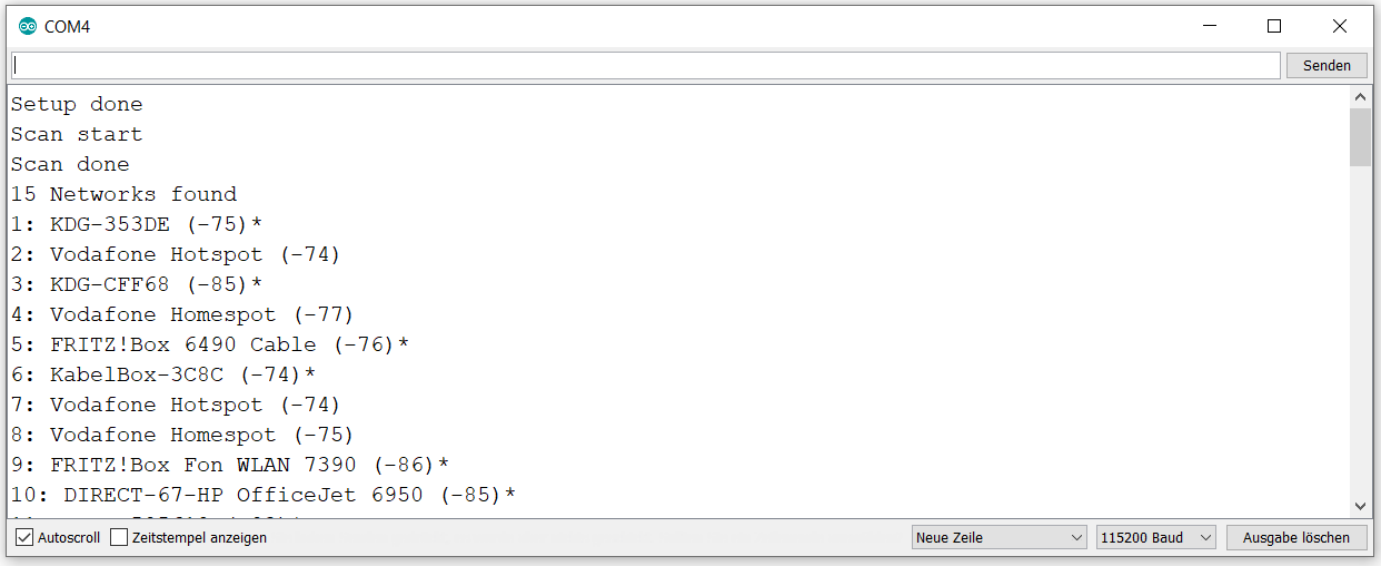




Sketch 42 Scan WiFi Networks (WLAN Netzwerke)

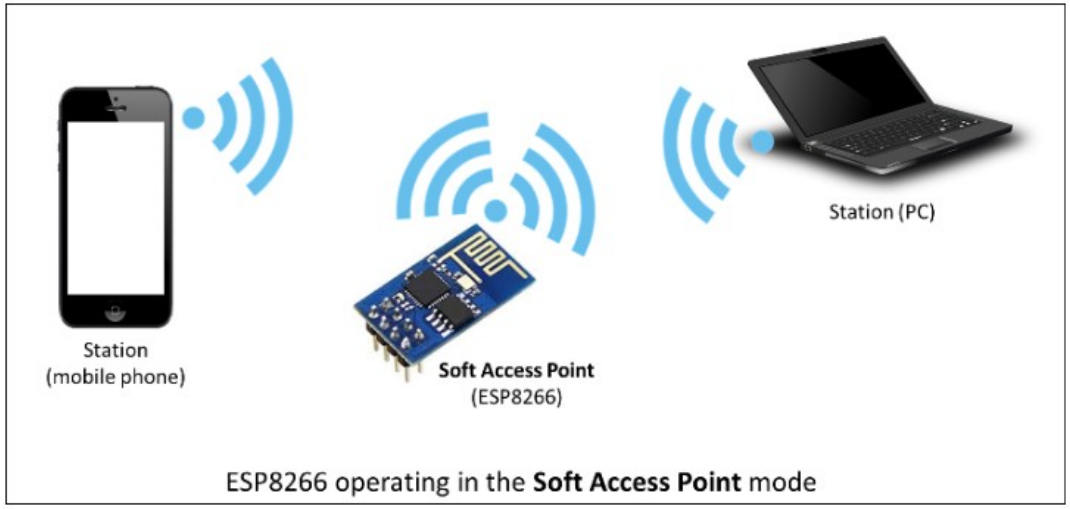
Es werden alle vom ESP erreichbaren WLAN-Netzwerke (WLAN-Access-Points) gesucht (gescannt).

Im Seriellen Monitor wird angezeigt:
Anzahl der WLAN-Netzwerke, deren Bezeichnung sowie Signalstärke.
Nach 10 Sekunden wird der Scan wiederholt.



(Beachte: Baudrate des Seriellen Monitors muss identisch zu derjenigen im Sketch sein)

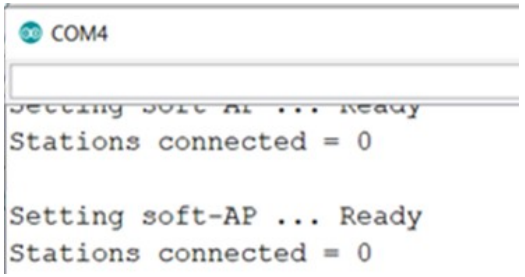
Sketch 43 ESP Soft Access Point



ESP8266 operating in the **Soft Access Point** mode

Der ESP8266 wird als Soft Access Point konfiguriert. Wi-Fi Stations (Clients) können sich zu ihm verbinden.

Nach Start des Sketches zeigt der Serielle Monitor vorerst:



Siehe auch:
https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/soft-access-point-examples.html

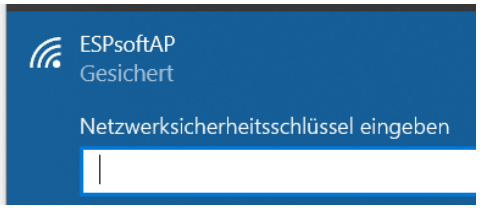
Zur Kommunikation des ESP über WiFi wird das Bibliotheksprogramm "ESP8266WiFi" verwendet.
Es wird bei der Installation des Programmpakets für den ESP mit vom Internet geladen und liegt auf:
.../portable\packages\esp8266\hardware\esp8266\2.5.0\libraries\ESP8266WiFi */

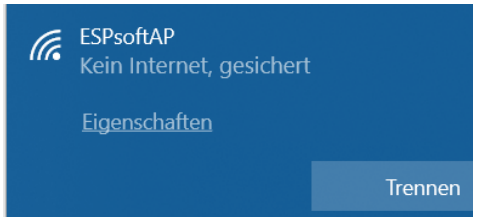
Sketch 43 ESP Soft Access Point

Wenn für Smartphone oder Computer WLAN aktiviert ist, sollte jetzt in der Liste der verfügbaren WLAN-Netzwerke mit erscheinen:

Nach Anklicken, Verbinden und Eingabe des Schlüssels:
(am Dell WLAN-Modul eingesteckt)

Siehe auch:
[Behandlung WLAN am Computer]

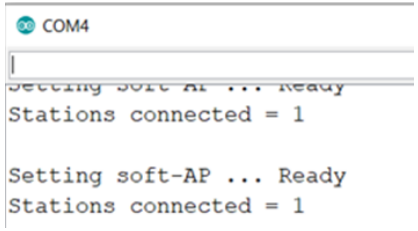




Rechter Mausklick führt zu “Eigenschaften”:

SSID:	ESPsoftAP
Protokoll:	802.11g
Sicherheitstyp:	WPA2-Personal
Netzfrequenzbereich:	2,4 GHz
Netzwerkkanal:	1
Verbindungslokale IPv6-Adresse:	fe80::67:2141:c502:dbf7%7
IPv4-Adresse:	192.168.4.2
IPv4-DNS-Server:	192.168.4.1

Der Serielle Monitor zeigt jetzt:

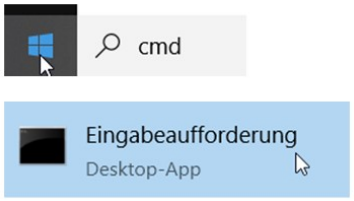


Bei Eingabe dieser Adresse [http://192.168.4.2] in einen Webbrowser kommt aber keine Anzeige (no response).

Sketch 43 ESP Soft Access Point - Ping

Ping ist ein Diagnose-Werkzeug, mit dem überprüft werden kann, ob ein bestimmter Access Point AP in einem Netzwerk erreichbar ist.
Man kann auch die Zeitspanne zwischen dem Aussenden eines Paketes zu diesem AP und dem Empfangen eines daraufhin unmittelbar zurückgeschickten Antwortpaketes sehen (= Paketumlaufzeit, *Round trip time* oder *RTT*).
Das Programm wird üblicherweise als Konsolenbefehl ausgeführt.

Wir prüfen, ob der ESP8266 Soft Access Point aktiviert ist:



dann in die Kommandozeile eingeben: ping 192.168.4.1

ESP8266 Soft Access Point aktiviert:

```
Eingabeaufforderung
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Wolfgang>ping 192.168.4.1

Ping wird ausgeführt für 192.168.4.1 mit 32 Bytes Daten:
Antwort von 192.168.4.1: Bytes=32 Zeit=45ms TTL=128
Antwort von 192.168.4.1: Bytes=32 Zeit=106ms TTL=128
Antwort von 192.168.4.1: Bytes=32 Zeit=548ms TTL=128
Antwort von 192.168.4.1: Bytes=32 Zeit=2ms TTL=128

Ping-Statistik für 192.168.4.1:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 2ms, Maximum = 548ms, Mittelwert = 175ms
```

ESP8266 Soft Access Point nicht aktiviert

```
Eingabeaufforderung
Microsoft Windows [Version 10.0.17763.437]
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Wolfgang>ping 192.168.4.1

Ping wird ausgeführt für 192.168.4.1 mit 32 Bytes Daten:
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.

Ping-Statistik für 192.168.4.1:
    Pakete: Gesendet = 4, Empfangen = 0, Verloren = 4
    (100% Verlust),
```

Sketch 44 ESP Soft Access Point und Webserver

Wie Sketch 43:

Der ESP8266 wird als Soft Access Point konfiguriert. Wi-Fi Stations (Clients) können sich zu ihm verbinden.

Wenn für SmartPhones oder Computer WLAN aktiviert ist, sollten sie jetzt diesen Soft AP anzeigen: ESPsoftAP

Hier wird nun noch ein Webserver programmiert.

Mit dem Bibliotheksprogramm (library) `<ESP8266WebServer.h>` wird auf dem ESP8266 ein HTTP-Webserver installiert.

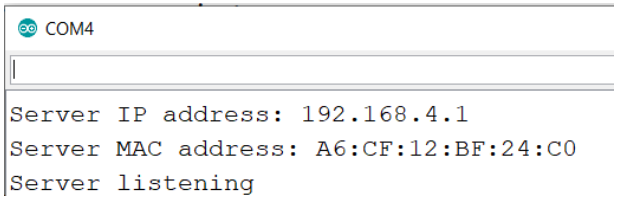
Der ESP8266 kann nun Daten senden zu einer verbundenen Station (auch Client genannt), z.B zu einem Computer.

Diese Daten können in einem Webbrowser (z.B. edge, chrome) angezeigt werden.

Dazu muß im Computer die Webseite http://192.168.4.1 aufgerufen werden.

Der ESP hat standardmäßig die IP-Adresse 192.168.4.1

Sie wird im Sketch 44 auch im Serial Monitor angezeigt.

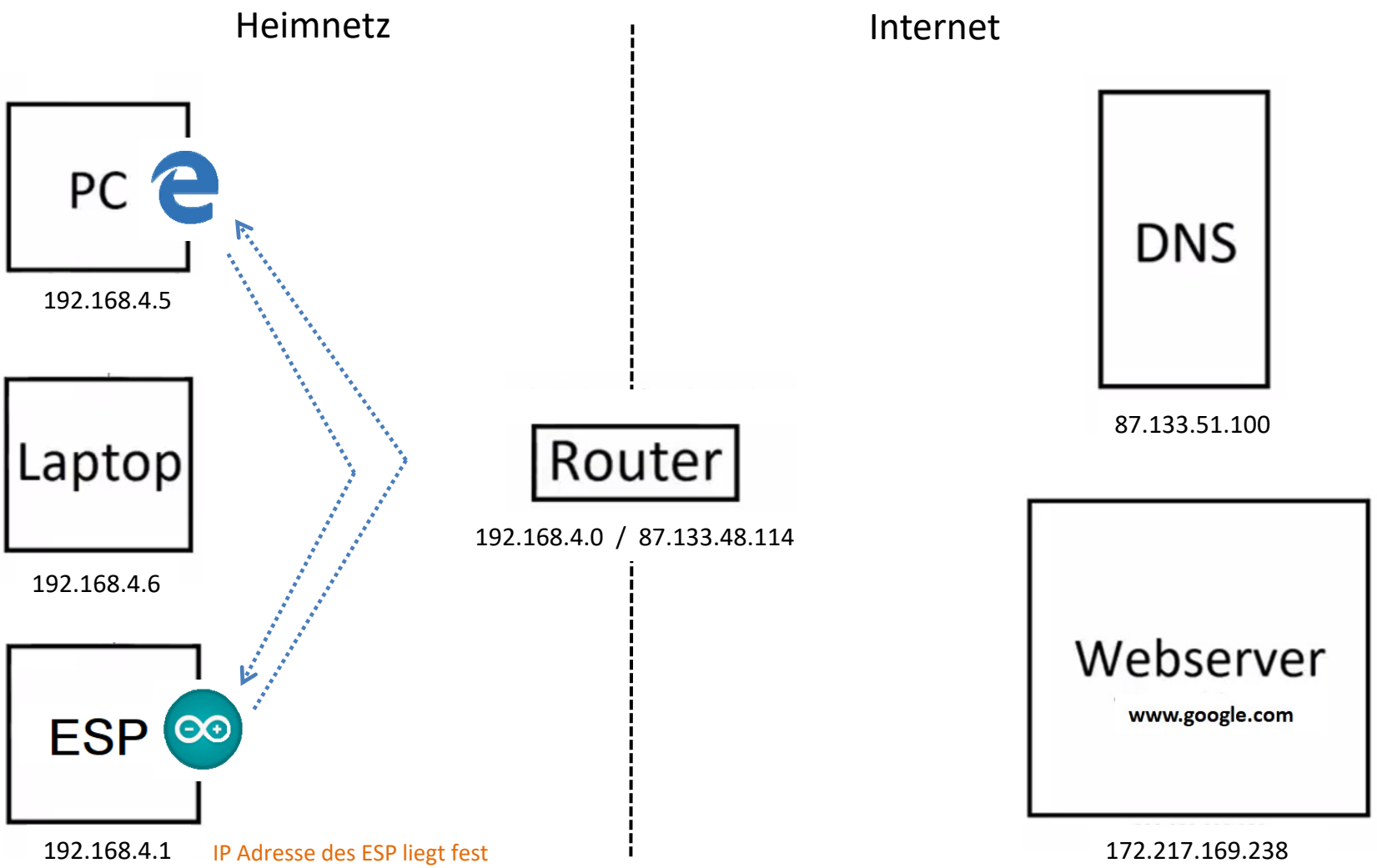


Die Kommmunikation vom ESP zum Computer ist per WLAN, aber noch nicht über das Internet !

Siehe auch:

https://techtutorialsx.com/2017/04/25/esp8266-setting-an-access-point/

Sketch 46 LED schalten über WLAN des ESP, im Heimnetz



Sketch 46 LED schalten über WLAN des ESP

Die auf dem ESP befindliche LED (auch verbunden mit GPIO 16 bzw. D0) wird von einem Computer/Smartphone ein/aus geschaltet.

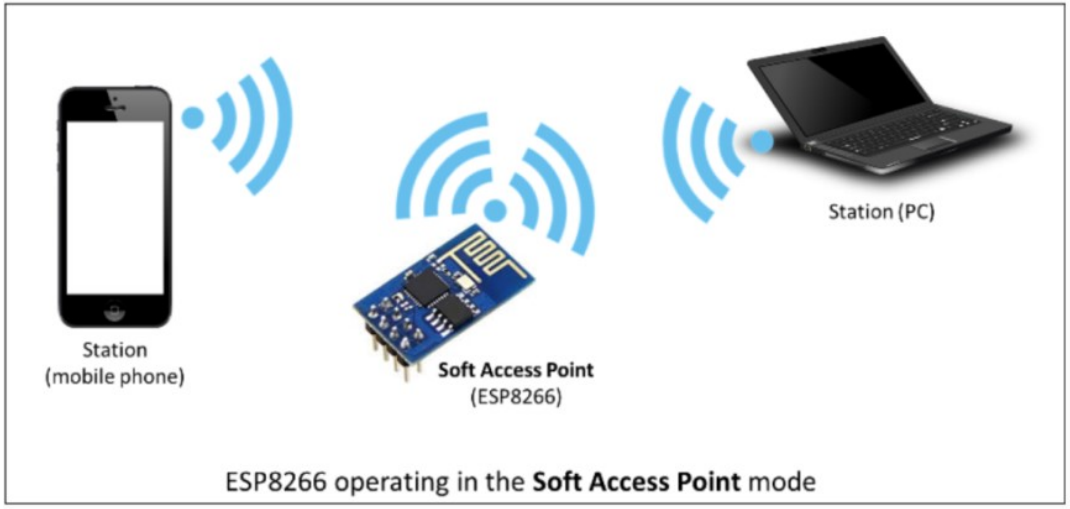
Das wird am Computer/Smartphone in einem Webbrowser ausgeführt, mit den Buttons „Schalte LED Ein" und „Schalte LED Aus".
Im Browser wird der Schalt-Status auch angezeigt: „LED STATUS: Ein“ bzw. „LED STATUS: Aus“

Der ESP fungiert als Server, Computer/Smartphone fungiert als Client.

Der Client stellt mit der Betätigung der zwei Buttons eine Anfrage (einen Request) an den Server.

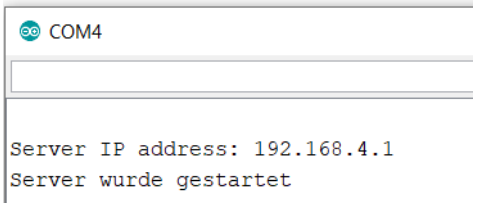
Der ESP ist als Access Point (AP) programmiert, er erzeugt ein eigenes WLAN-Netz.

Es erfolgt keine Verbindung über das Internet, deshalb wird der AP als „Soft AP“ bezeichnet..

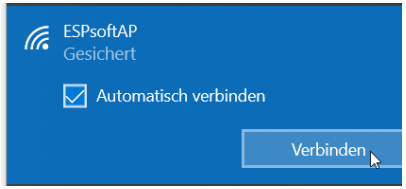


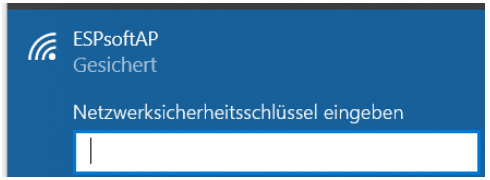
Sketch 46 LED schalten über WLAN des ESP

Nach Start des Sketches kommt folgende Anzeige im Serial Monitor (einmalig, ohne Wiederholungen):

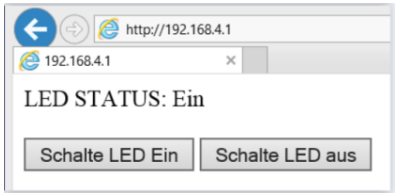


Am Computer oder Smartphone nach Anklicken, Verbinden und Eingabe des Schlüssels:
(am Dell WLAN-Modul eingesteckt)
Siehe auch:
Behandlung WLAN am Computer

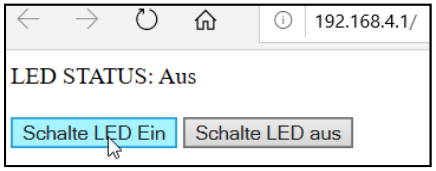




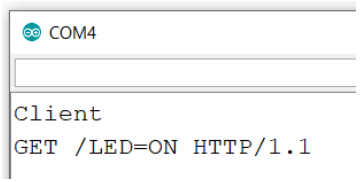
Im Webbrowser des Computers/Smartphones muß die IP-Adresse des ESP aufgerufen werden: **http://192.168.4.1**
Dann erscheint die Webseite mit den zwei Buttons.



Wenn ein Button am Computer/Smartphone gedrückt wird:



Es erscheint im Serial Monitor:



Sketch 46 Erläuterungen zum HTML Code

client.println("

");
Break (Umbruch, nächste Zeile). Einfügen von zwei Leerzeilen auf die Webseite.

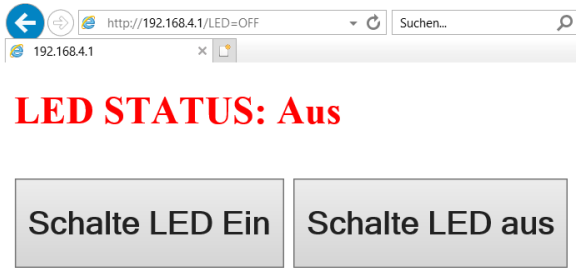
client.print("<h1 style='color:red'>");

client.println("<button>Schalte LED Ein</button>");
Auf die Webseite wird ein Hyperlink geschrieben, der in einem Button untergebracht ist, zu der Web-Unterseite
http://192.168.4.1/LED=ON
Anklicken des Buttons [Schalte LED aus] erzeugt einen Hyperlink zu dieser Web-Unterseite.

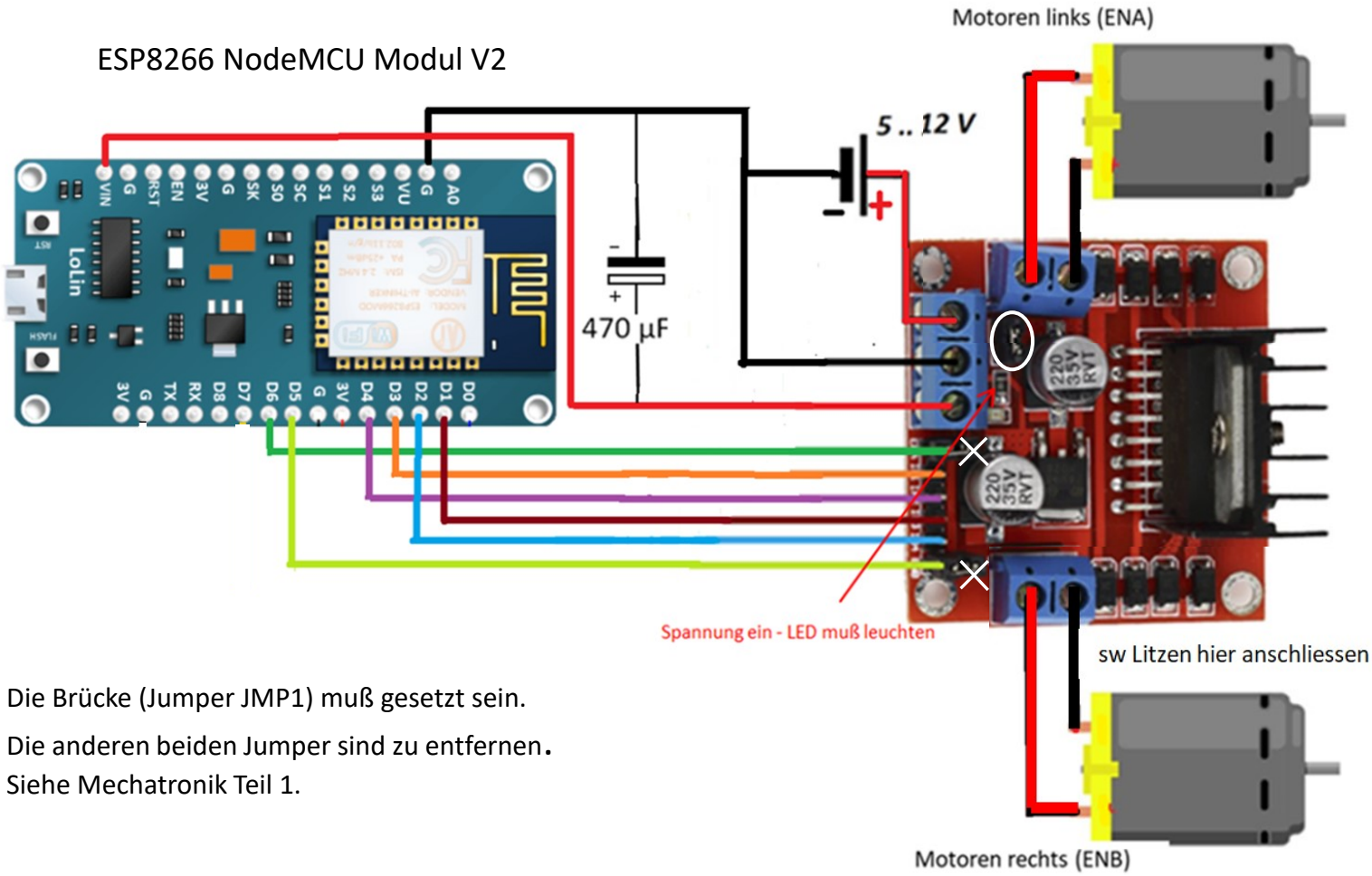
Wie müßte man den Hyperlink ohne Button schreiben ?

Siehe auch:
https://www.html-seminar.de/interne-links.htm

Was müßte man ändern, um die Webseite so darzustellen ?



Sketch 47 Auto fahren über WLAN des ESP



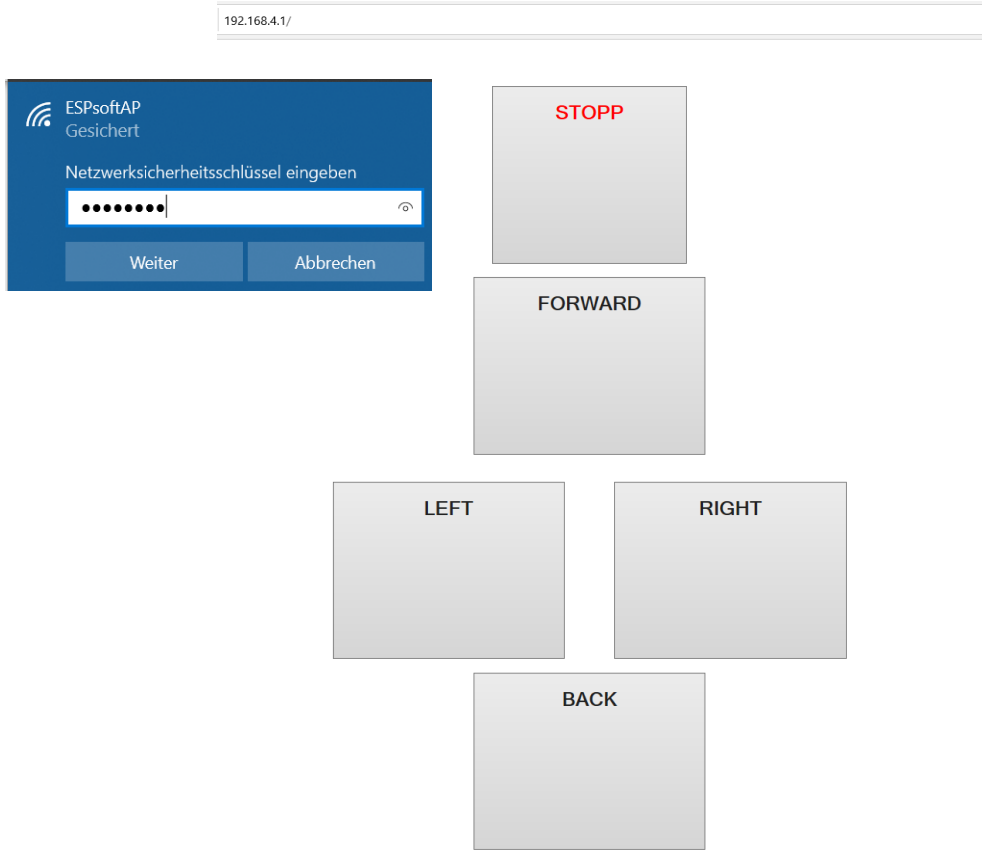
Die Brücke (Jumper JMP1) muß gesetzt sein.

Die anderen beiden Jumper sind zu entfernen.
Siehe Mechatronik Teil 1.

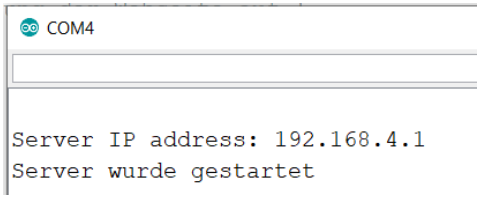
ESP	D6	D3	D4	D1	D2	D5
L298	ENA	INA1	INA2	INA3	INA4	ENB

Sketch 47 Auto fahren über WLAN des ESP

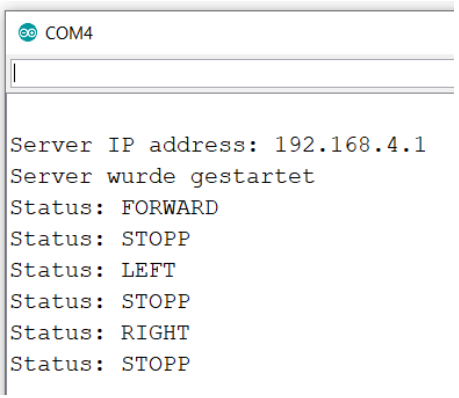
Das Auto wird in einem Webbrowser mit Buttons gesteuert, die vom Sketch des ESP erzeugt werden.
Der ESP arbeitet im AP-Mode(Access Point).
Im Webbrowser muss die IP-Adresse des ESP aufgerufen werden: **http://192.168.4.1** (wenn nicht mit WiFiConfig() geändert wurde).
Dann erscheint die Webseite mit den Buttons und auch noch eine Anzeige zum Fahrstatus.



Kurz nach Start des Sketches zeigt der Serielle Monitor an:



Und nach Betätigung von Buttons:



Sketch 32 Greifer öffnen und schliessen

Greifer mit Servo MG996R

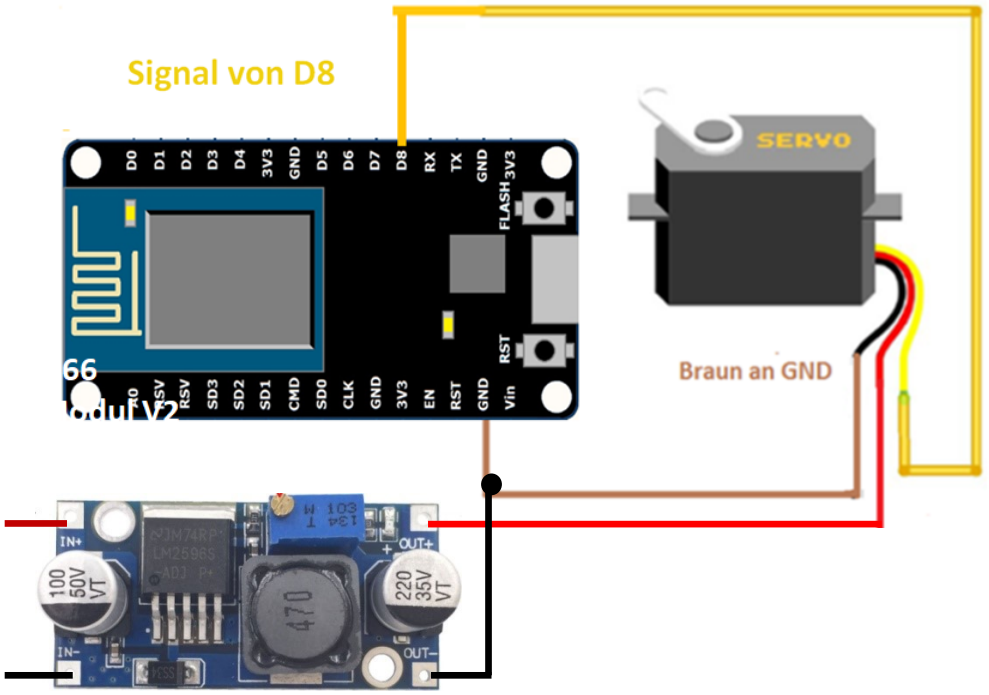
Sketch 32 Greifer öffnen und schliessen

Der ESP erhält die Spannung vom Ausgang 5V des L298N.

Wenn der Servo ebenfalls von dieser Spannung gespeist wird, kann es zu Fehlfunktionen des ESP kommen durch kurzzeitige Spannungseinbrüche.

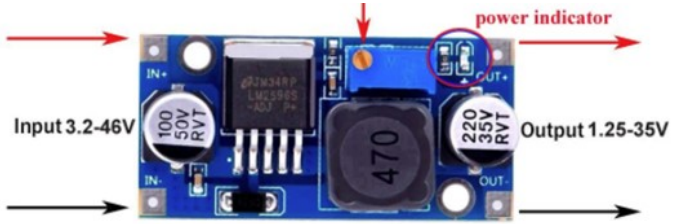
Deshalb erhält der Servo die Spannung über eine separate Stromversorgung (Einstellung auf ca. 6V).

Vom Akku
9,6V (Leerlauf voll geladen >10,5V)

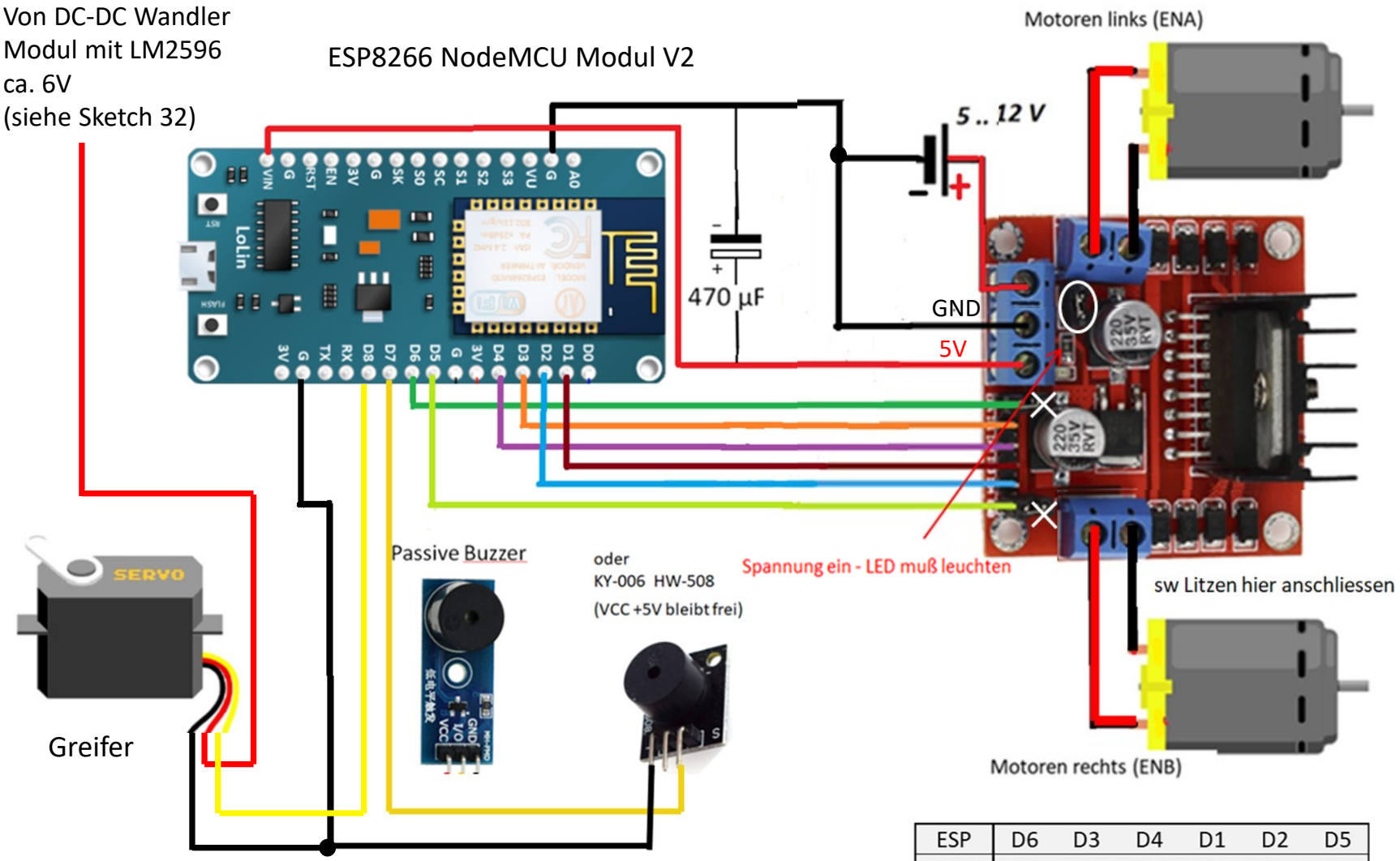


DC-DC Wandler Modul mit LM2596

Einstellung Ausgangsspannung ca. 6V

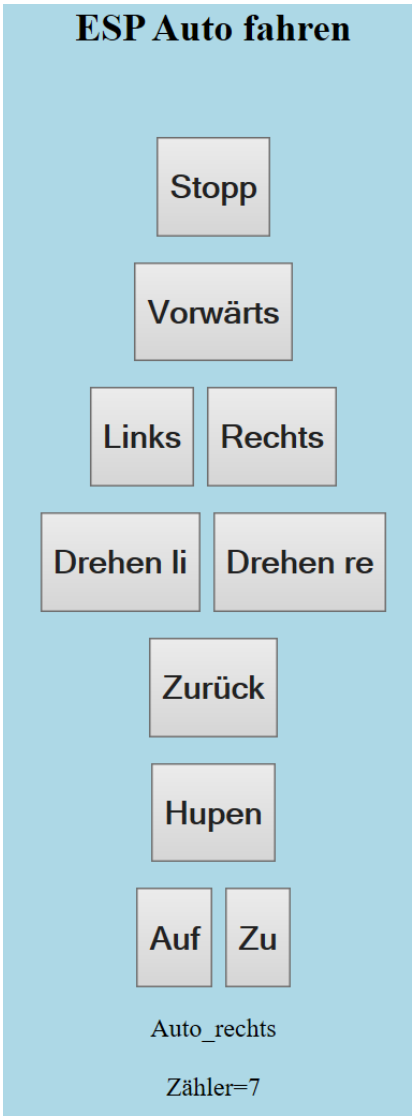


Sketch 48 Auto + Hupen + Greifer über WLAN des ESP



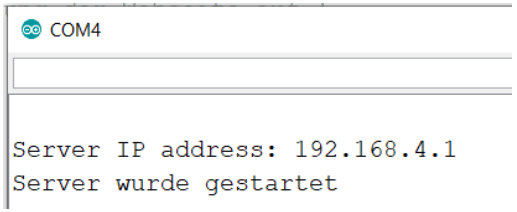
ESP	D6	D3	D4	D1	D2	D5
L298	ENA	INA1	INA2	INA3	INA4	ENB

Sketch 48 Auto + Hupen + Greifer über WLAN des ESP



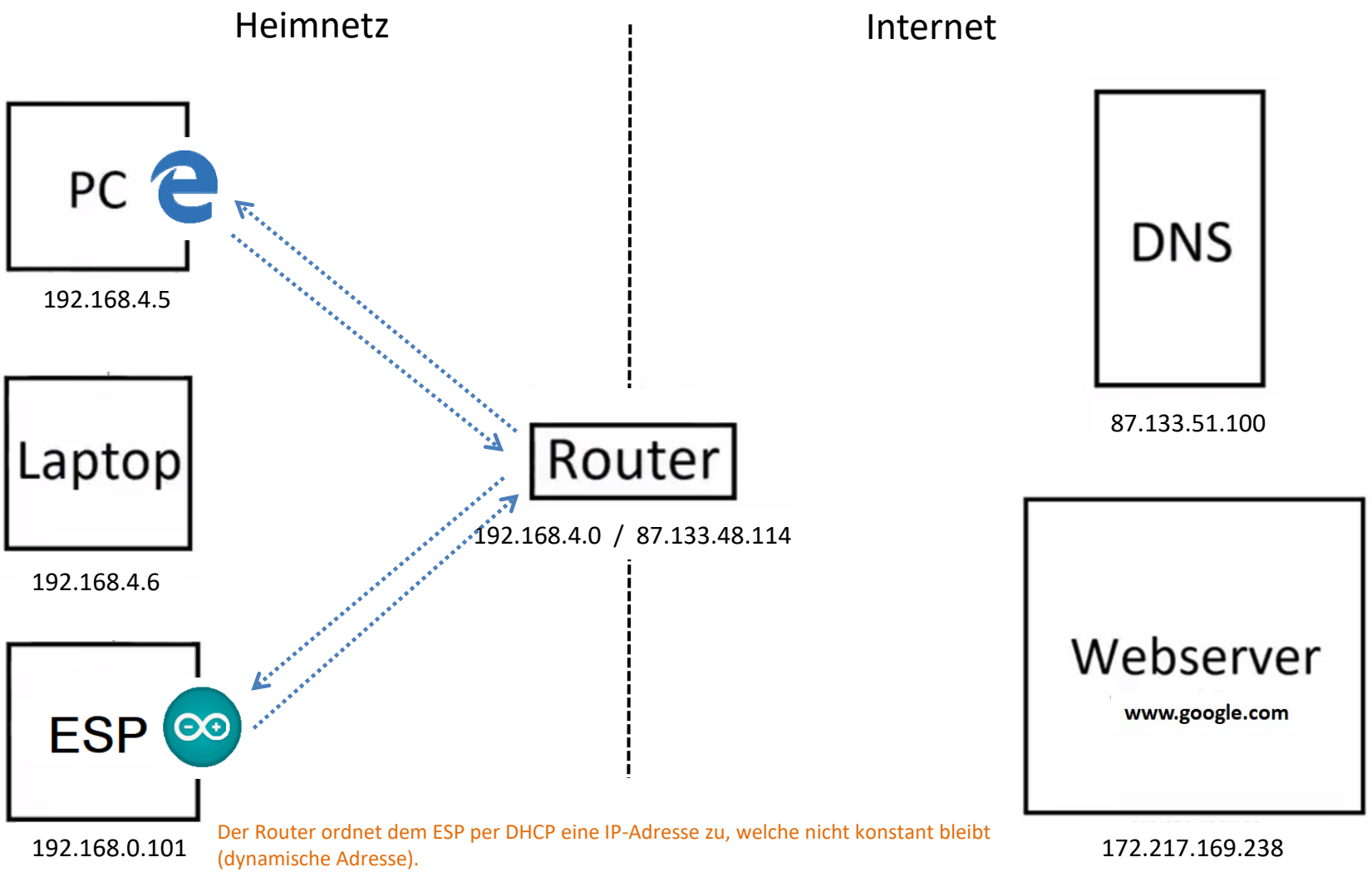
Der ESP arbeitet wiederum im AP-Mode(Access Point).
Dieser Sketch erzeugt zusätzliche Buttons zur Steuerung von Hupe und Greifer.
Es ist hier ein stabiler Webserver programmiert, bei Fehlern kann die Webseite wieder aktualisiert (refresh) werden.
Im Webbrowser muss die IP-Adresse des ESP aufgerufen werden:
http://192.168.4.1 (wenn nicht mit WiFiConfig() geändert wurde).

Kurz nach Start des Sketches zeigt der Serielle Monitor an:



Und nach Betätigung von Buttons:

Sketch 49 LED schalten über WLAN des Routers , im Heimnetz



Der Router ordnet dem ESP per DHCP eine IP-Adresse zu, welche nicht konstant bleibt (dynamische Adresse).
Sie kann aber im Sketch per WiFiconfig geändert werden, bleibt dann auch konstant (statische Adresse).

Sketch 49 LED schalten über WLAN des Routers

Die auf dem ESP befindliche LED (auch verbunden mit GPIO 16 bzw. D0) wird von einem Computer/Smartphone ein/aus geschaltet.

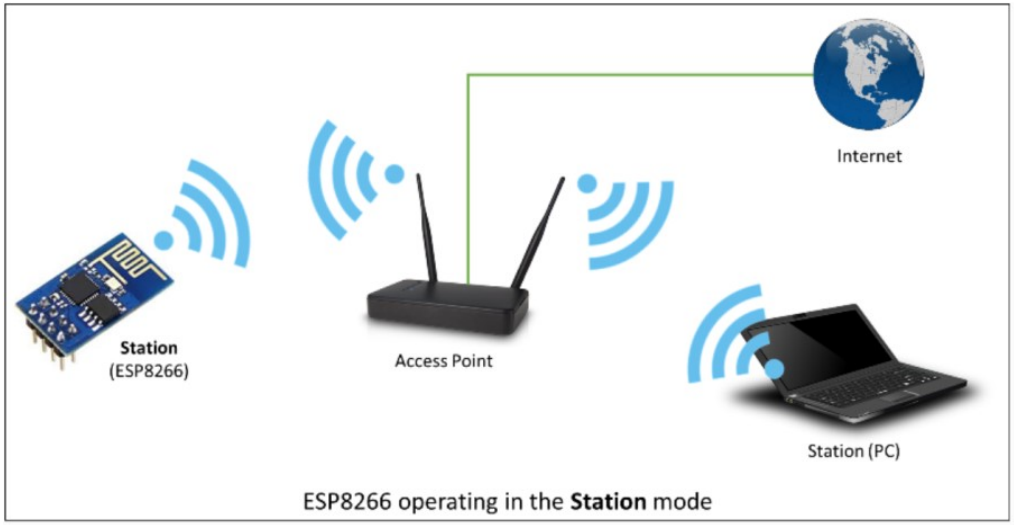
Das wird am Computer/Smartphone in einem Webbrowser ausgeführt, mit den Buttons „Schalte LED Ein" und „Schalte LED Aus".
Im Browser wird der Schalt-Status auch angezeigt: „LED STATUS: Ein“ bzw. „LED STATUS: Aus“

Der ESP fungiert als Server, Computer/Smartphone fungiert als Client.
Der Client stellt mit der Betätigung der zwei Buttons eine Anfrage (einen Request) an den Server.

ESP als auch Computer/Smartphone sind mit dem WLAN-Netz des Routers verbunden, es erfolgt aber keine Verbindung über das Internet.
Der ESP loggt sich in ein bestehendes WLAN ein, erzeugt vom Router, und bezieht von dort per DHCP eine IP Adresse (bisher 192.168.9.222).

Als Router kann auch ein Smartphone verwendet werden, wenn ein HotSpot aktiviert ist:

Siehe:
Hotspot erzeugen (Tethering)



ESP8266 operating in the **Station** mode

Sketch 49 LED schalten über WLAN des Routers

Nach Upload des Sketches (bzw. Reset) kommt folgende Anzeige im Serial Monitor:

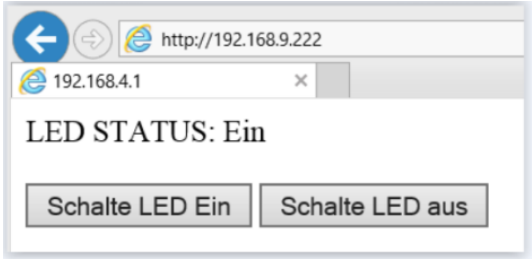
```
COM4
.....
ESP ist verbunden mit dem WLAN-Netzwerk des Routers
Server(ESP)wurde gestartet
Der Router hat dem ESP die IP-Adresse zugeordnet: 192.168.0.222
Client hat sich verbunden
GET /LED=OFF HTTP/1.1
```

Wenn ein Button am Computer/Smartphone gedrückt wird, erscheint „Client hat sich...“:

Am Computer/Smartphone muß nach dem WLAN-Netzwerk des Routers (nicht des ESP) gesucht und auf „Verbinden“ gedrückt werden.

Im Webbrowser des Computers/Smartphons muß die IP-Adresse des ESP aufgerufen werden:
http://192.168.0.222

Dann erscheint die Webseite mit den zwei Buttons.



indexOf() https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/indexof/

Sketch 49 LED schalten über WLAN des Routers

WiFi.status() https://www.arduino.cc/en/Reference/WiFiStatus

Returns if ESP is connected to a WLAN network.

WiFi.status() ==WL_CONNECTED	connected
WiFi.status() != WL_CONNECTED	not connected

WiFi.begin()

Calling it will instruct the ESP to switch to the station mode and connect to the last used access point based on configuration saved in flash memory

```
WiFi.begin();
WiFi.begin(ssid);
WiFi.begin(ssid, password);
```

connect() https://www.arduino.cc/en/Reference/ClientConnect

Connects to a specified IP address and port. The return value indicates success or failure.

Wichtig:

available() https://www.arduino.cc/en/Reference/WiFiServerAvailable

WiFiServer() https://www.arduino.cc/en/Reference/WiFiServer

Sketch 49 LED schalten über WLAN des Routers, feste IP

Man kann mit dem Befehl „WiFi.config“ bewirken, daß der Router dem ESP eine feste (statische) IP-Adresse zuordnet, die sich nicht ändert.

Es werden auch die DNS, Gateway, and Subnet- Adressen des ESP geändert.

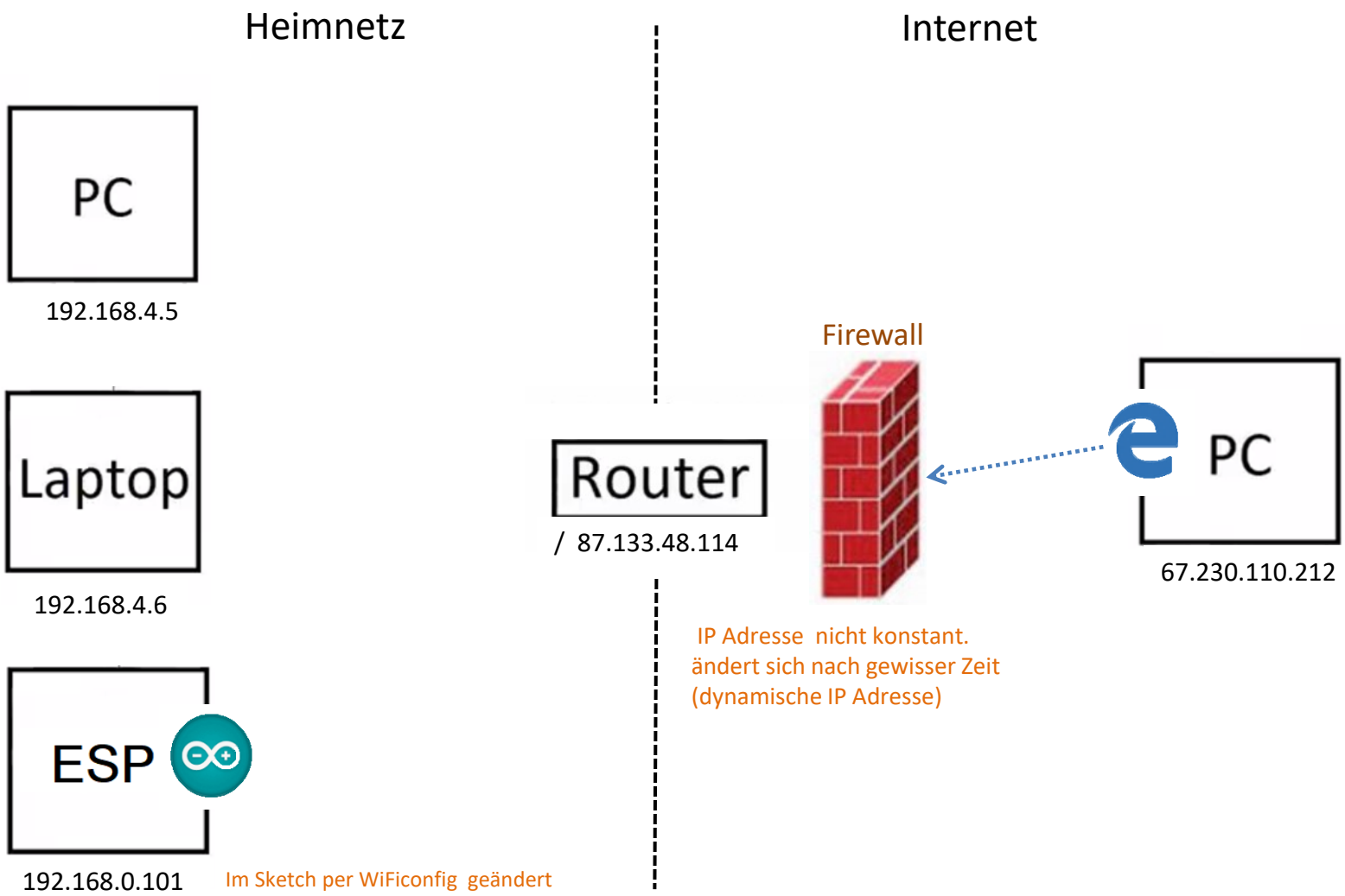
In den Sketch wird eingefügt:

```
/*Feste IP-Adresse:*/
IPAddress ip(192,168,0,101);
IPAddress gateway(192,168,0,1);
IPAddress subnet(255,255,255,0);
WiFi.config(ip, gateway, subnet);
```

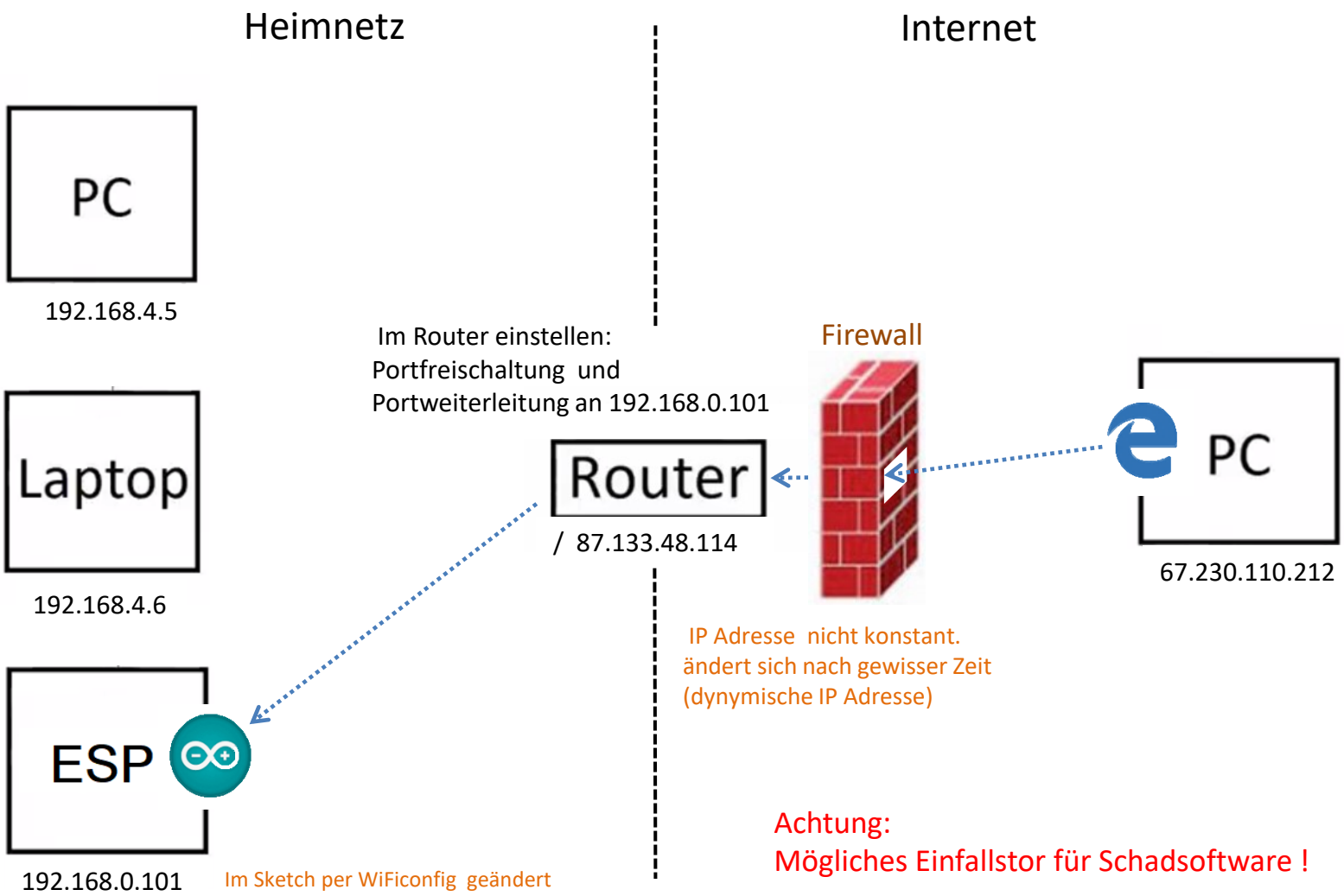
Im Browser muß nun die IP-Adresse http://192.168.0.101 aufgerufen werden.

WiFi.config()	https://www.arduino.cc/en/Reference/WiFiConfig

Auto steuern über Internet – Hindernis Firewall



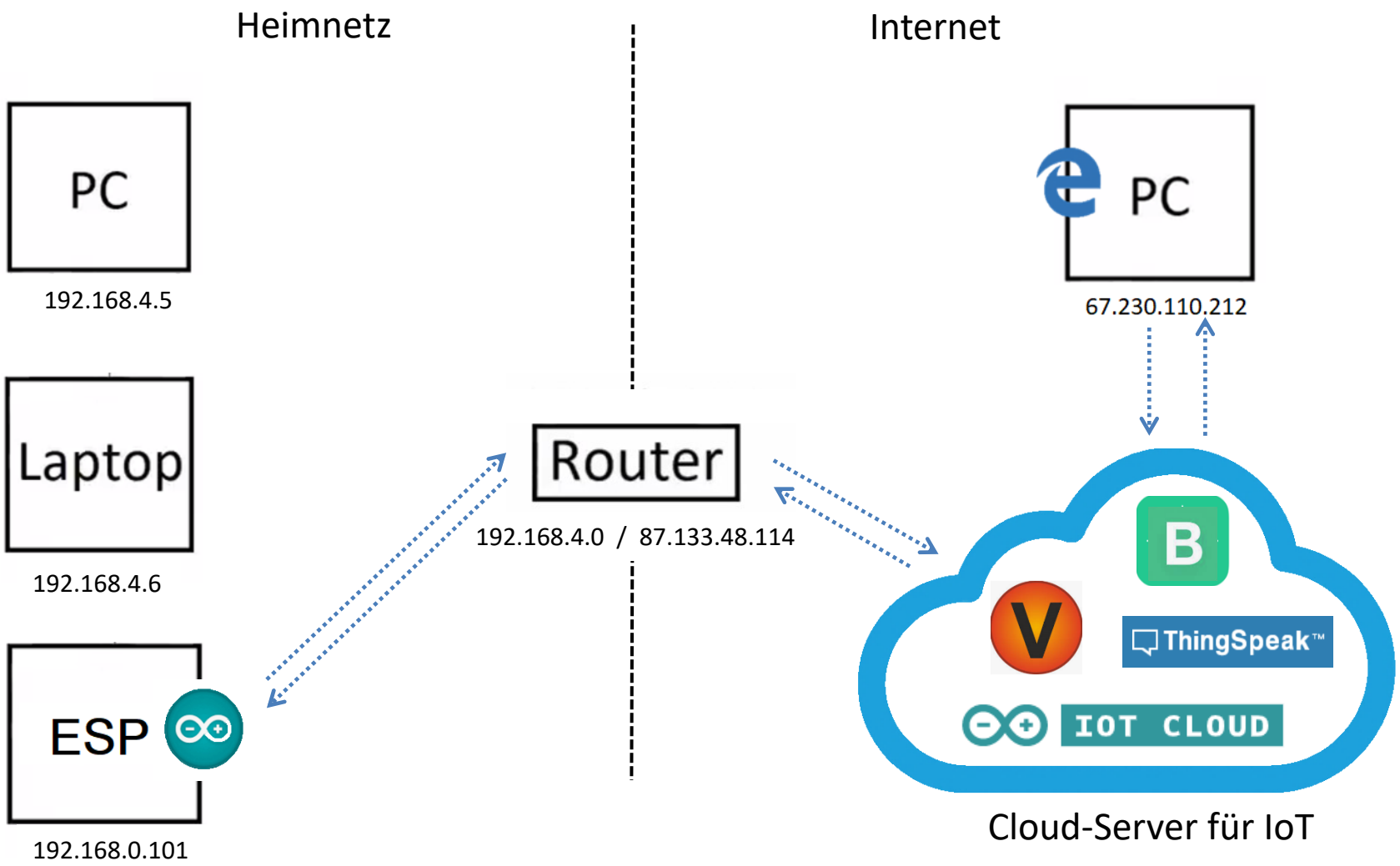
Auto steuern über Internet - Portfreischaltung



https://www.youtube.com/watch?v=Rl86Km6cjBk

https://www.youtube.com/watch?v=GyKHUARJJbs

Auto steuern über Internet – Cloud-Server für IoT



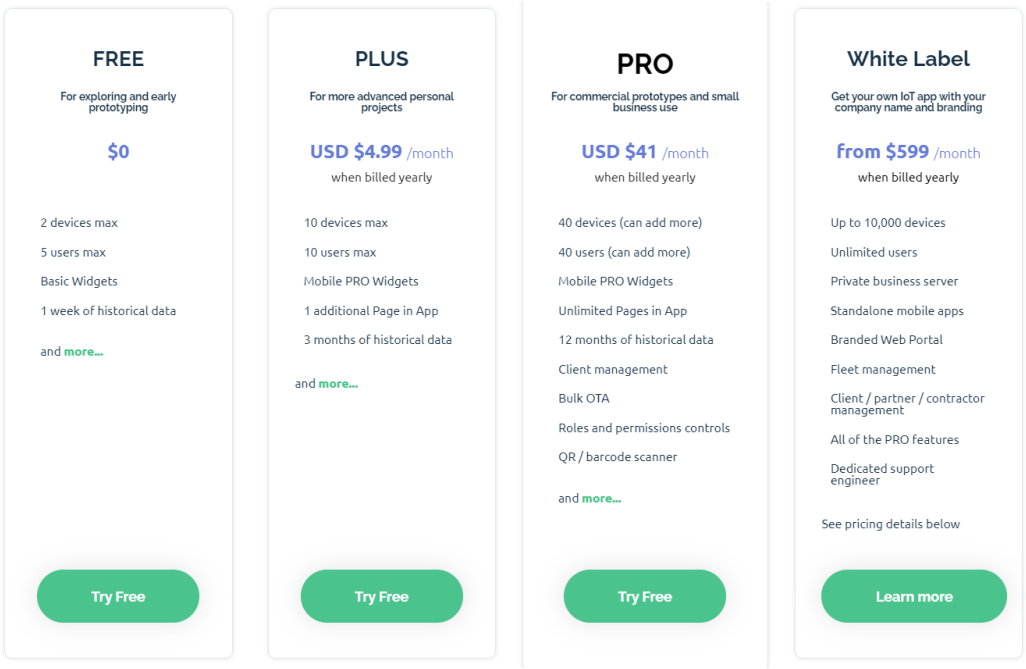
IoT-Plattform „Blynk“

https://blynk.io/

Blynk ist ein Software-Unternehmen, gegründet 2014, mit Hauptquartier in New York und Niederlassung in Berlin.

Es werden eine Software-Plattform und Server zur Verfügung gestellt für die Kommunikation über das Internet, besonders für Anwendungen bei IoT.

Die Software wurde 2021
völlig neu konzipiert.
Zu den Kosten der Nutzung siehe
https://blynk.io/pricing .



Wir nutzen die Option FREE.
Damit können allerdings nur 2 Geräte gesteuert werden und die Anzahl an „Widgets“ (Betätigungselemente wie Button, Slider oder Joystick) ist sehr begrenzt.

Wir steuern das Ein-Ausschalten einer LED (Sketch 50) sowie das Fahren des Autos (Sketch 52).

Siehe auch:
https://tutorial.cytron.io/2021/10/21/getting-started-with-the-new-blynk-iot-platform-using-maker-uno/

Kommunikation über Blynk

Blynk-Server

Verbindung ins Internet über LTE

Verbindung ins Internet über Festnetz (Router) oder Smartphone

Smartphone mit Blynk-App

Router oder Smartphone (am Auto)

Verbindung zum ESP über WLAN

Registrierung auf der Blynk –Website
Passwort anlegen
Download App BLYNK

Smartphone auf Auto anbringen:
Mobile HotSpot (Tethering) aktivieren
WLAN-Namen+WLAN Passwort wird angezeigt

Bibliotheksprogramm „Blynk“ in ESP laden.

In Sketch eintragen:
WLAN-Namen und WLAN-Passwort

Template ID und Auth Token von Blynk

Start und Registrierung für „Blynk“ am Computer

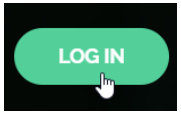
Zuerst wird eine Nutzer-Registrierung erforderlich.

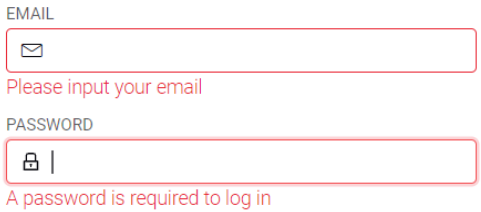
https://blynk.io/ dann Button 

Es öffnet sich eine Webseite zur Anmeldung: https://blynk.cloud/dashboard/register

Eingabe der Email-Adresse sowie Anlegen eines Passwortes.

Künftig einwählen wiederum bei https://blynk.io/ und mit Login-Button oben rechts:





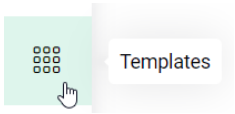
Die Einstellungen bei Blynk (Anlegen der Projekte) werden auf der Website durchgeführt.

Die spätere Kommunikation zwischen ESP und Blynk-Server kann ebenfalls über die Website erfolgen (am Computer, auch als „Console“ bezeichnet, als auch am Smartphone) oder über eine Smartphone-App.

Sketch 50 - ein Template anlegen für Schalten einer LED

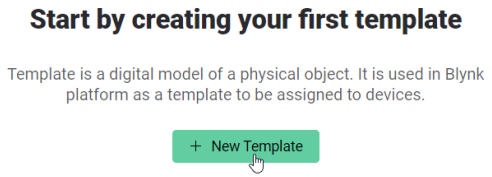
Wir wollen über den Blynk-Server die auf dem ESP8266 befindliche LED ein- und ausschalten.
Folgende Schritte müssen auf der Website ausgeführt werden:

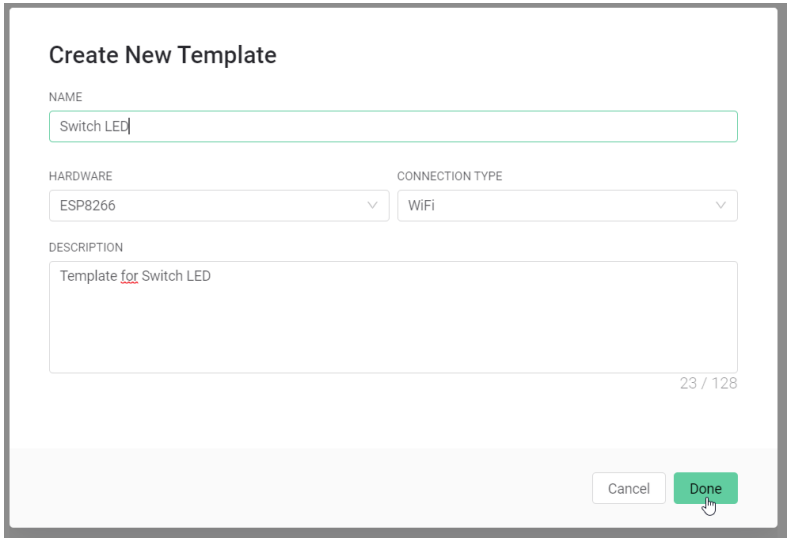
Anklicken links oben:



Löschen des vorhandenen „Quickstart Template“.

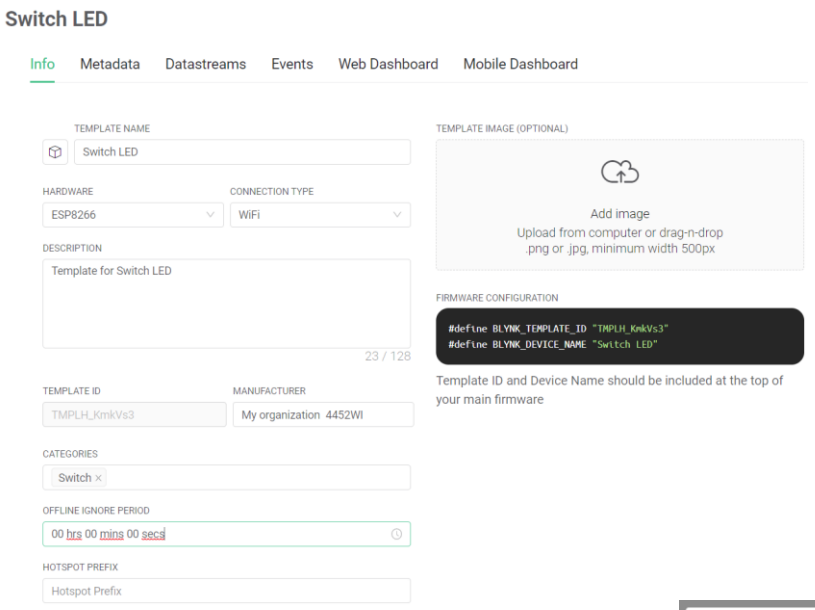
Neues Template erstellen:





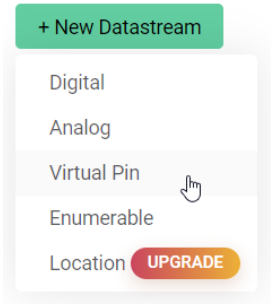
Sketch 50 - ein Template anlegen für Schalten einer LED

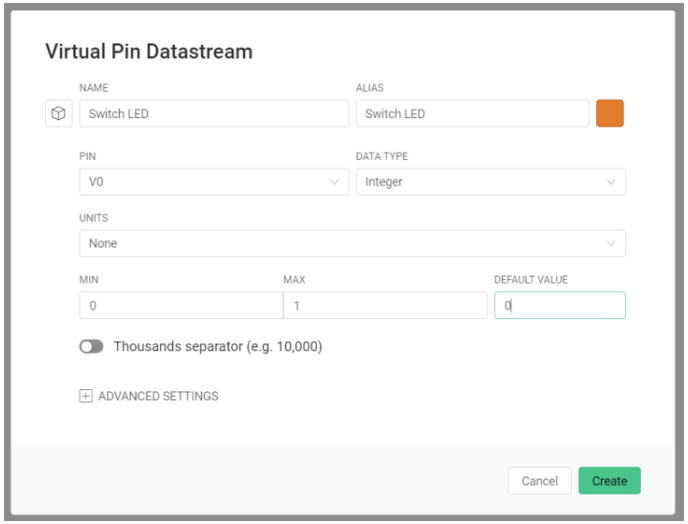
Es öffnet sich:



„Datastreams“ anklicken und wählen „Virtual Pin“:

(zu „Virtual Pins“ siehe Anhang)





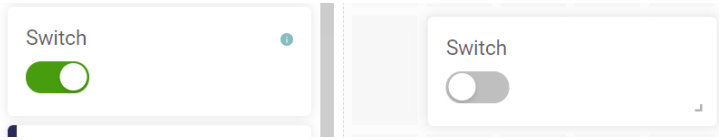
Sketch 50 - ein Template anlegen für Schalten einer LED

Es öffnet sich:

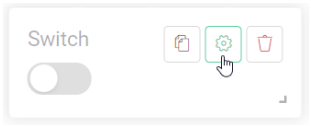
1 Datastream

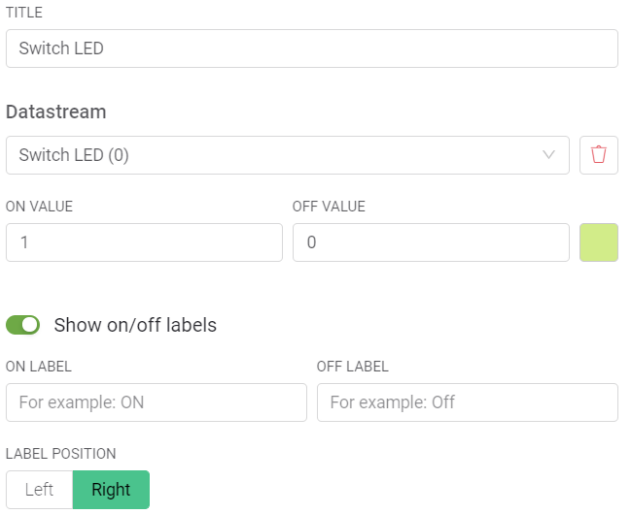
Id	Name	Color	Pin	Data Type	Units	Min	Max	Decimals	Default Value
1	Switch LED		0	Integer		0	1	--	0

„Web Dashboard“ anklicken.
Das Widget „Switch“ rüberziehen.

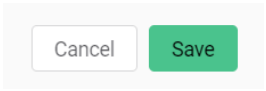


Auf Zahnrad klicken
und Settings vornehmen:



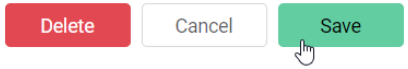


Button „Save“:

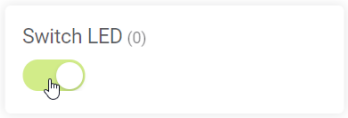


Sketch 50 - ein Template anlegen für Schalten einer LED

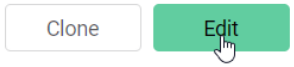
Nochmal Button „Save“ rechts oben:



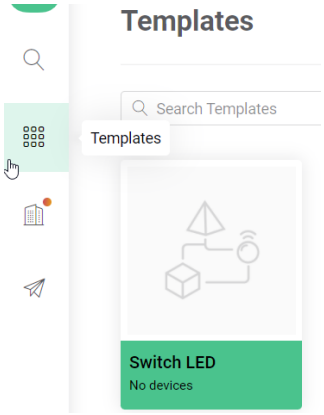
Jetzt ist das Web-Dashboard fertig :



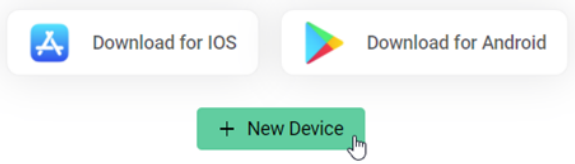
Mit Button „Edit“ rechts oben könnten Änderungen durchgeführt werden:



Bei Anklicken von „Templates“ erscheint das neu angelegte Template.
Allerdings ist noch kein „Device“ angelegt.

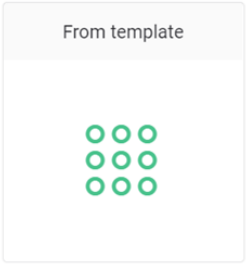


Jetzt oben links „Devices“ anklicken
und „New Device“ anklicken:

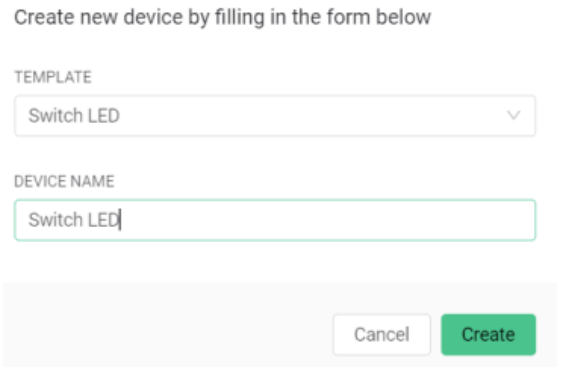


Sketch 50 - das Device anlegen für Schalten einer LED

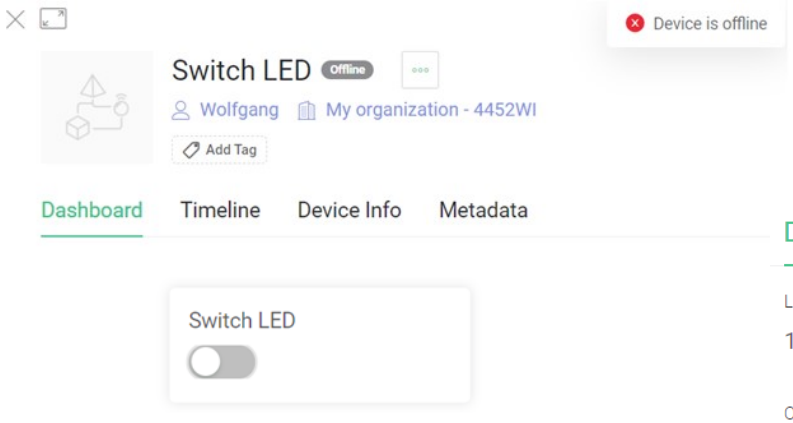
Anklicken „Device from Template“



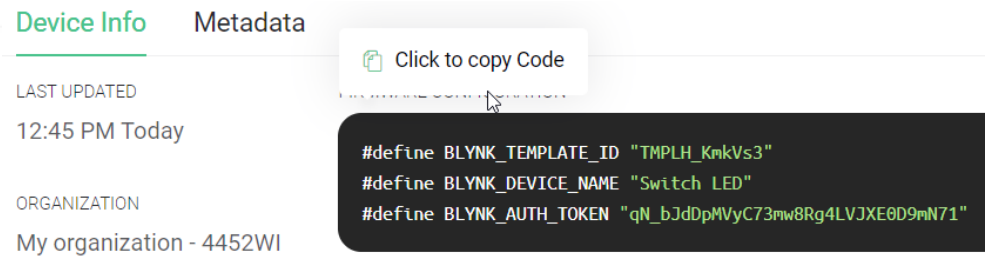
und eintragen:



Nach anklicken „Create“ erscheint:



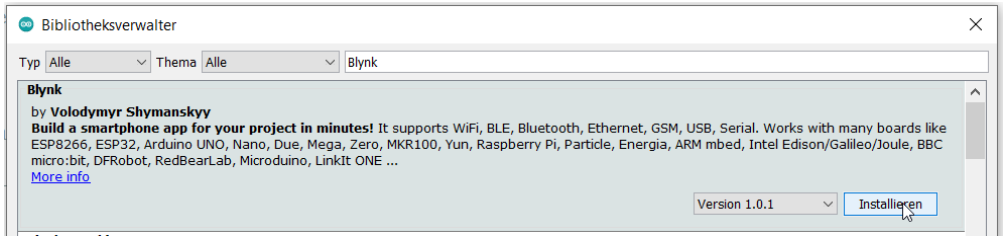
Diese Daten müssen in den Sketch kopiert werden:



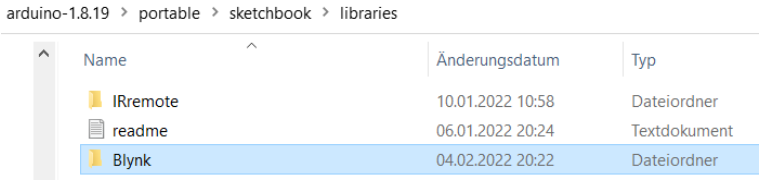
„Blynk“ ist bereit zur Datenübertragung – aber das „Device“ – der ESP8266 – ist offline.
Es muß noch ein Sketch erstellt werden und der ESP Verbindung zum Internet bekommen.

Sketch 50 - Bibliotheksprogramme in Arduino IDE

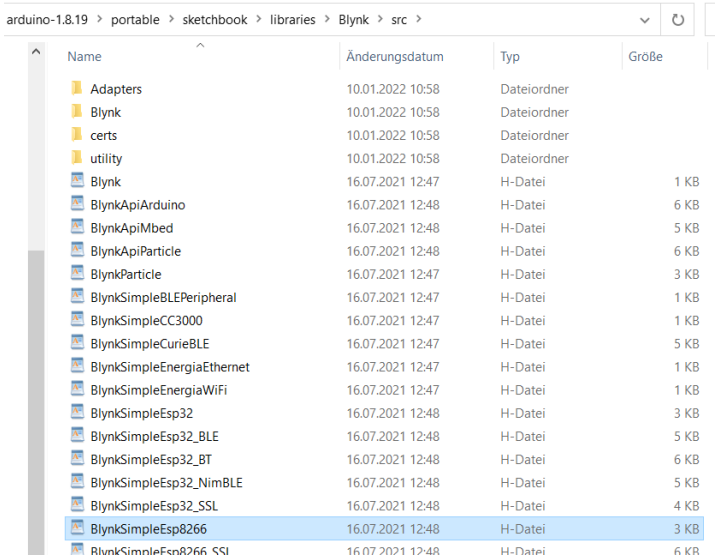
Das Bibliotheksprogramm „Blynk“ muß in die Arduino IDE geladen werden.



Das Bibliotheksprogramm befindet sich dann im Ordner:



Wir benötigen davon BlynkSimpleEsp8266.h
(im Sketch Zeile: #include <BlynkSimpleEsp8266.h>):



Sketch 50 - Hinweise zum Sketch

Siehe sketch_50_ESP_Blynk_LEDschalten

Wichtig:

In den Sketch müssen die vom Blynk-Server erhaltenen Daten (siehe weiter oben) eingetragen werden:

```
#define BLYNK_TEMPLATE_ID "TMPLzUnoXocZ"
#define BLYNK_DEVICE_NAME "Switch LED"
#define BLYNK_AUTH_TOKEN "8zyVVJ4PxNe_8yatLZky10Ecmf3jqac9"
```

Der ESP8266 wird mit dem Internet verbunden .

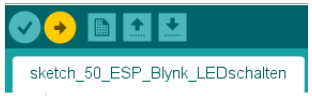
Das geschieht über ein Smartphone, das in der Nähe des ESP angeordnet ist und in dem Tethering (Bildung eines WLAN-HotSpot) aktiviert wurde.

SSID und Passwort müssen in den Sketch eingetragen werden.

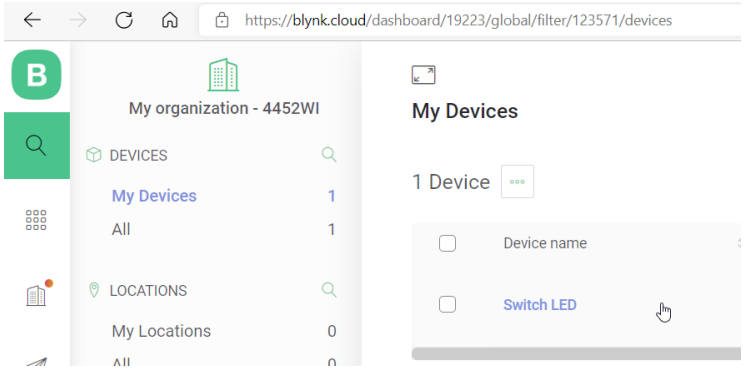
```
char ssid[] = "AndroidAP3232";     //SSID (Name) des Smartphone, das den HotSpot fuer den ESP bildet
char pass[] = "20c93d0xxxxx";      //WLAN-Passwort fuer den Zugriff auf das Smartphone
```

Sketch 50 - Starten

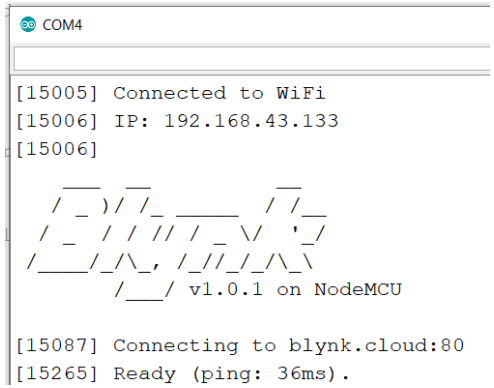
❶ Der Sketch wird in den ESP geladen.



❷ Am Computer Login zu Blynk und anklicken „Switch LED“:

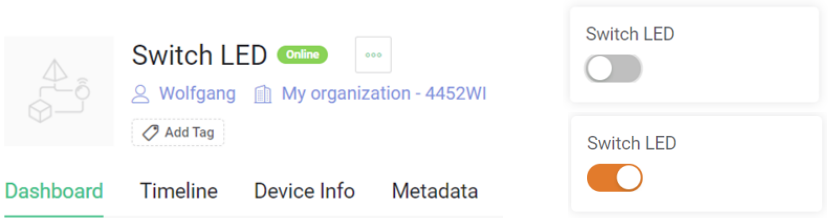


❸ Reset am ESP8266 – am Serial Monitor sollte erscheinen:

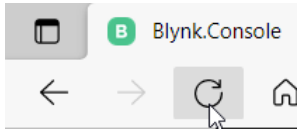


❹ Das Dashboard sollte jetzt anzeigen „Online“.

❺ Jetzt kann geschaltet werden.



Wenn die Funktion ausfällt –> Reset am ESP8266 oder Aktualisieren Website.



App „Blynk“ im Smartphone installieren

Blynk kann auch über eine App mit dem Smartphone ausgeführt werden.
Android OS version 4.2+ oder iOS version 9+.

Voraussetzung ist, daß die Registrierung bereits über die Website erfolgt war.

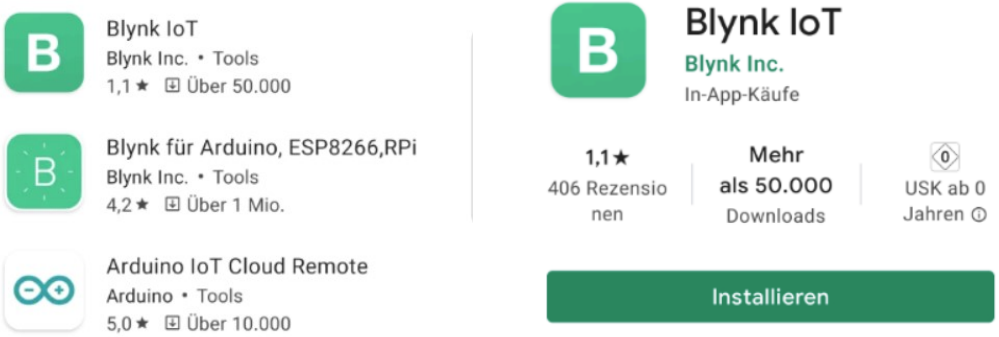
Download App „Blynk“

Download on the App Store

GET IT ON Google Play

Neue Version:

Alte Version:

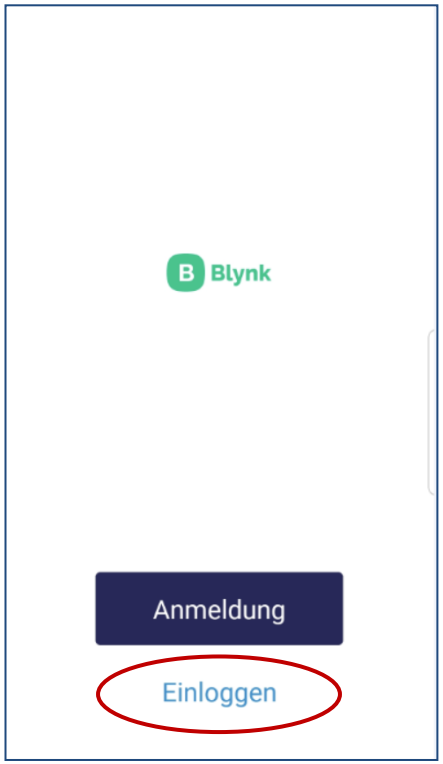


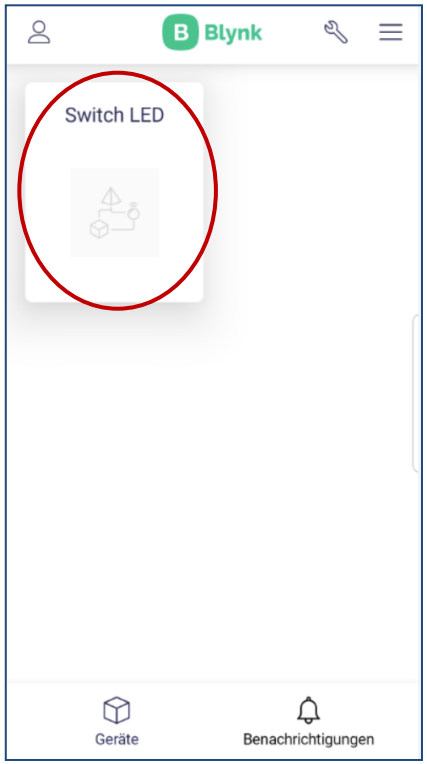
Icon im Smartphone:

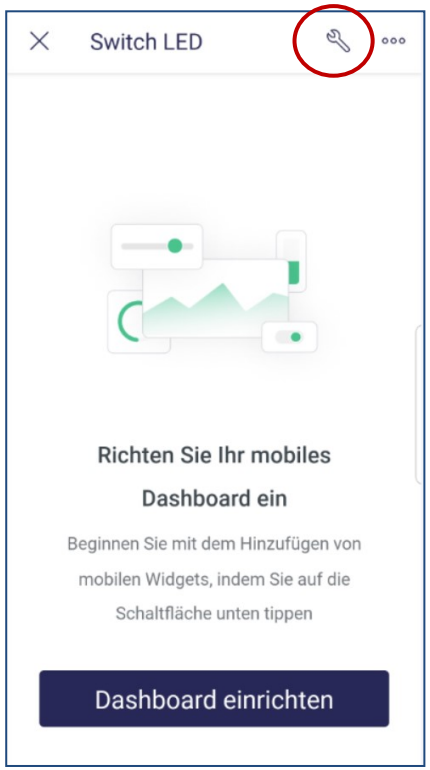
App „Blynk“ starten und „Switch LED“ einrichten

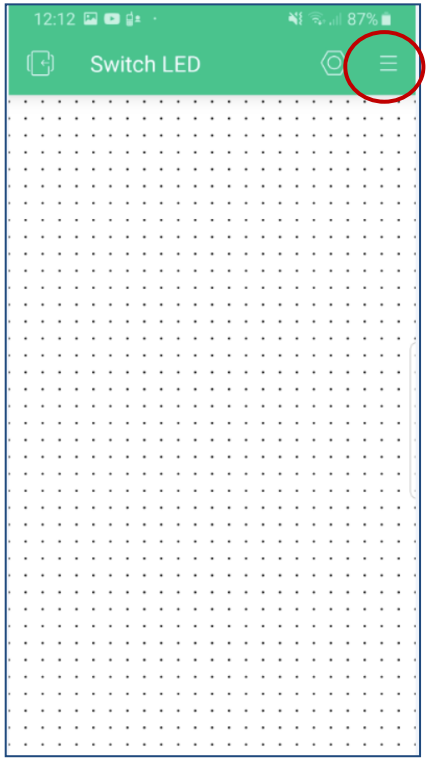
Einloggen (Zugangsdaten wurden bereits auf der Website angelegt)

Das Projekt „Switch LED“ wurde bereits auf der Website angelegt und erscheint auch in der App









App „Blynk“ starten und „Switch LED“ einrichten

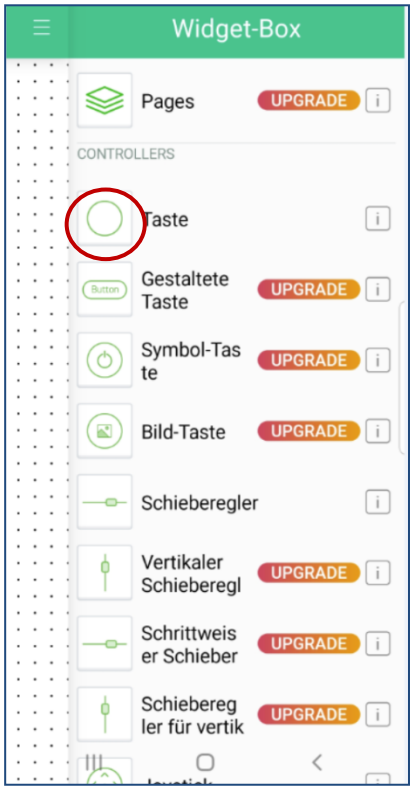
Widget „Taste“ anklicken

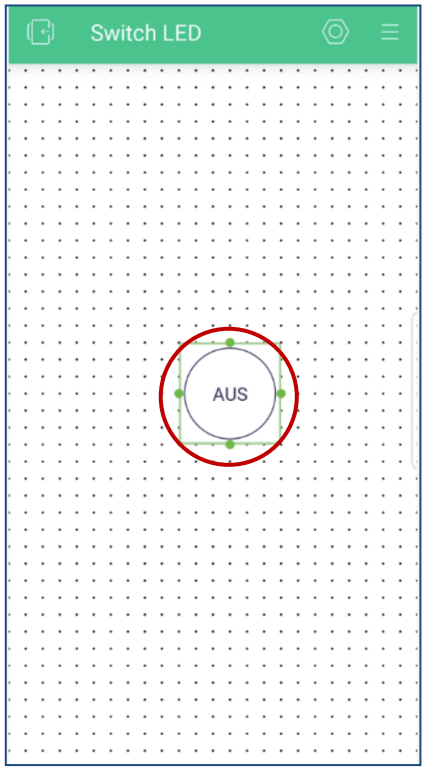
Auf die Taste klicken

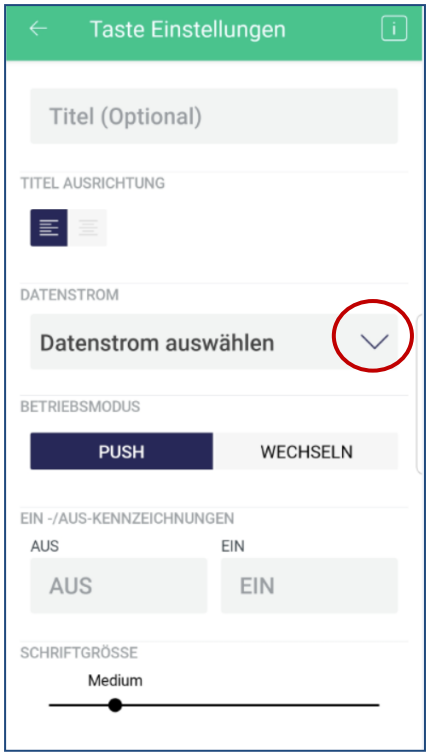
Datenstrom „Switch LED“ auswählen

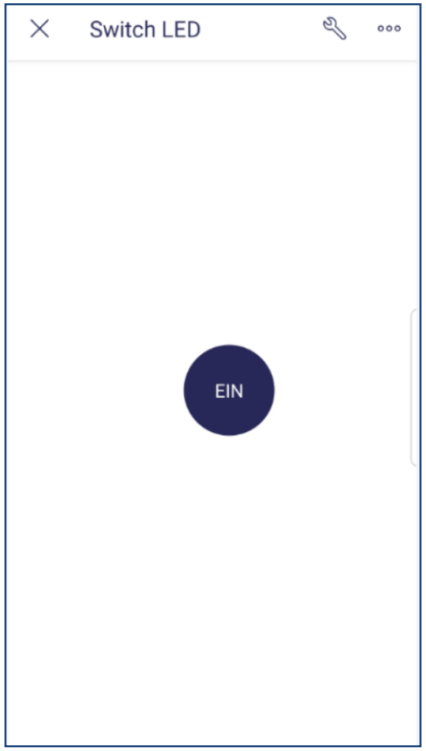
Aus „Entwickler-Mode“ zurück 

Jetzt ist die Funktion bereit.

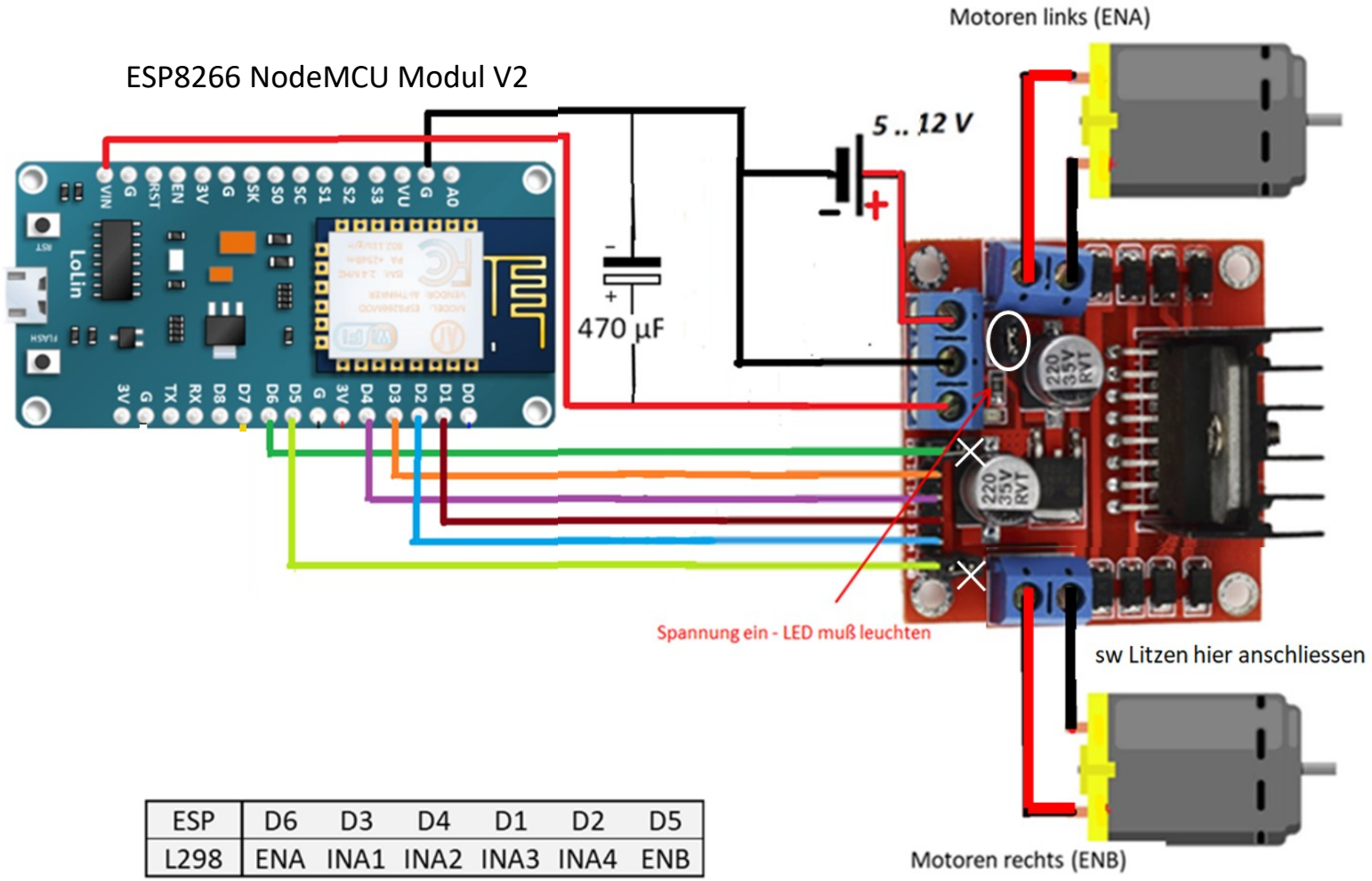








Sketch 52 Auto fahren über Blynk



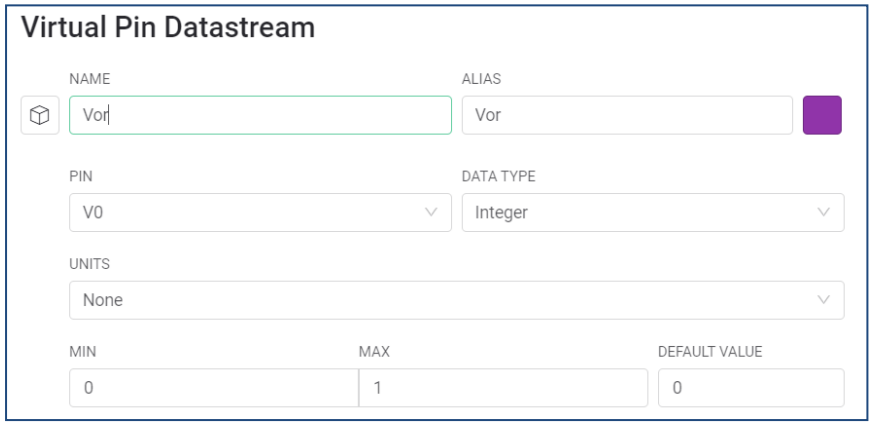
ESP	D6	D3	D4	D1	D2	D5
L298	ENA	INA1	INA2	INA3	INA4	ENB

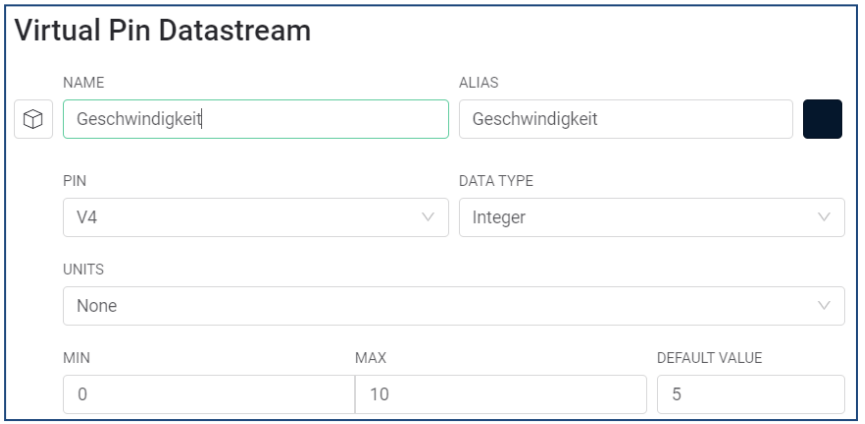
Sketch 52 Template und Device anlegen auf der Website

5 Datastreams anlegen für Buttons:
Virtual Pins V0, V1, V2, V3 und V5

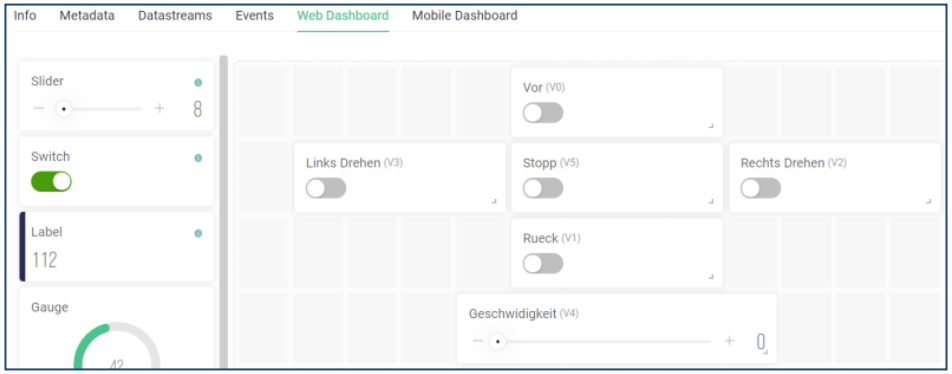
1 Datastream anlegen für Slider:
Virtual Pin V4

Bsp.: V0



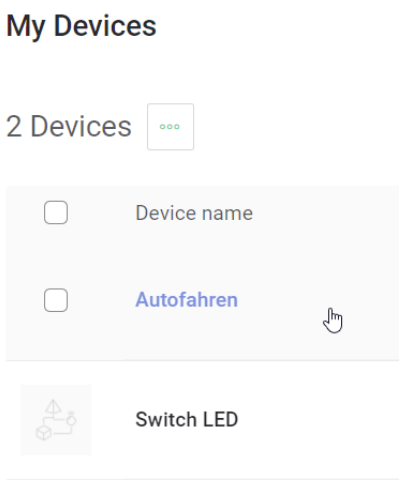


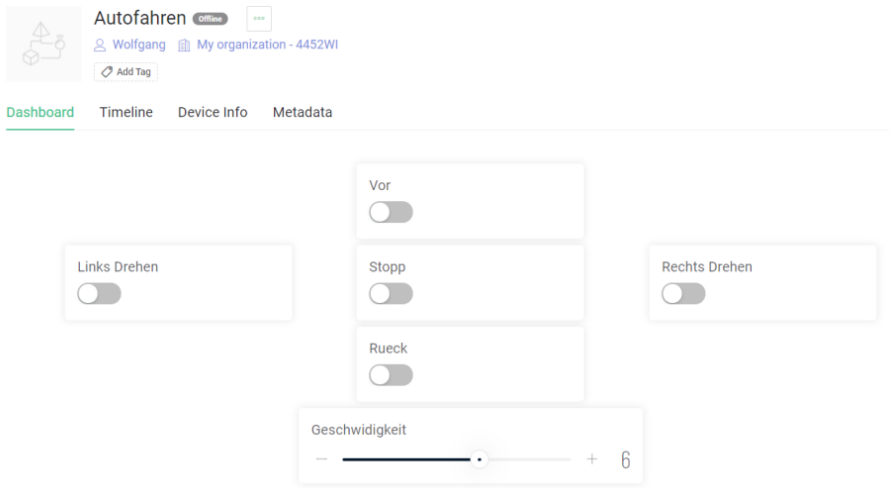
Dashboard anlegen



Sketch 52 Template und Device anlegen auf der Website

Device anlegen





Diese Daten in den Sketch kopieren:

FIRMWARE CONFIGURATION

```
#define BLYNK_TEMPLATE_ID "TMPLjly43WpC"
#define BLYNK_DEVICE_NAME "Autofahren"
#define BLYNK_AUTH_TOKEN "-c8PXoVIxpN4wSWzr16tb92D0c5OY1Y3"
```

Template ID, Device Name, and AuthToken should be declared at the very top of the firmware code.

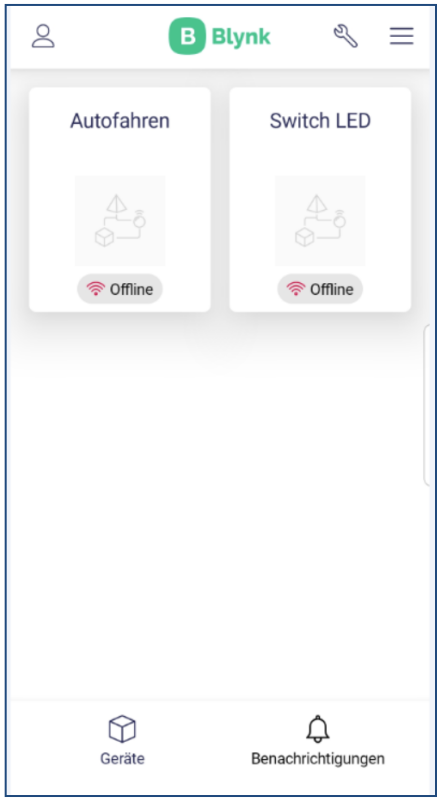
„Auto Fahren“ in der App anlegen

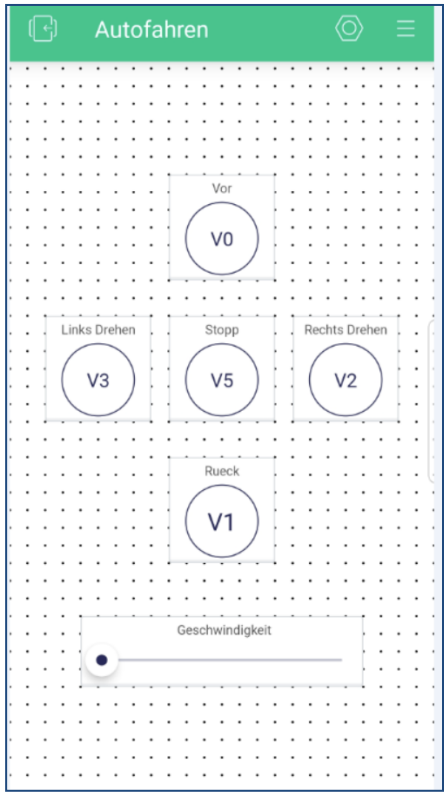
Das Projekt „Auto fahren“ wurde bereits auf der Website angelegt und erscheint auch in der App

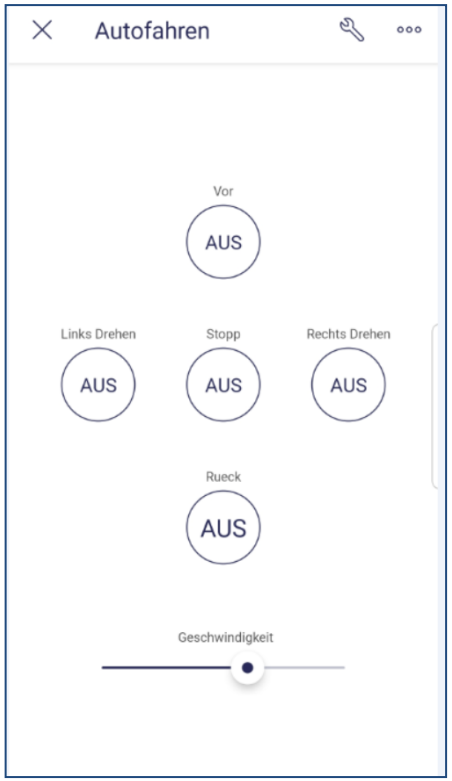
Widgets:
5 Tasten und einen Schieberegler anlegen

Aus „Entwickler-Mode“ zurück

Jetzt ist die Funktion bereit.







Sketch 52 Auto fahren über Blynk - Kamera über WhatsApp

Wir wollen am Auto eine Kamera anbringen.
Dazu nutzen wir WhatsApp.

Am ESP wird wiederum Sketch 52 gestartet.

Smartphone Nr.1:
Blynk wird gestartet und in geteilter Bildschirmansicht dargestellt („split screen“).

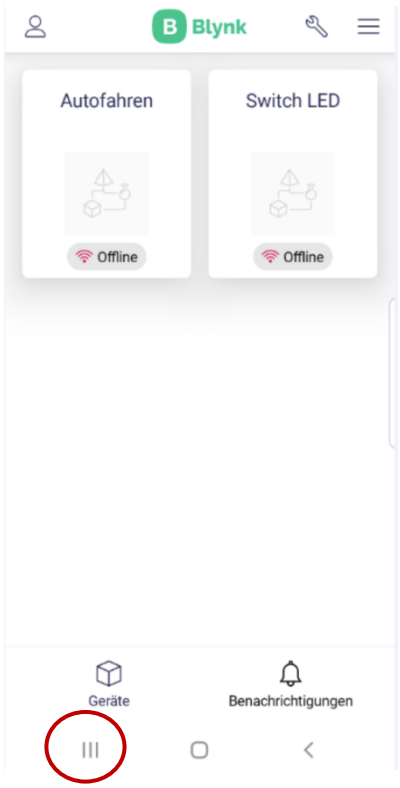
Smartphone Nr.2 (am Auto, bildet den HotSpot für den ESP):
Startet einen WhatsApp Videoanruf zu Smartphone Nr.1.

Smartphone Nr.1:
Nimmt den Anruf an.
Jetzt kann das Auto gesteuert und die Fahrt mit der Kamera verfolgt werden.

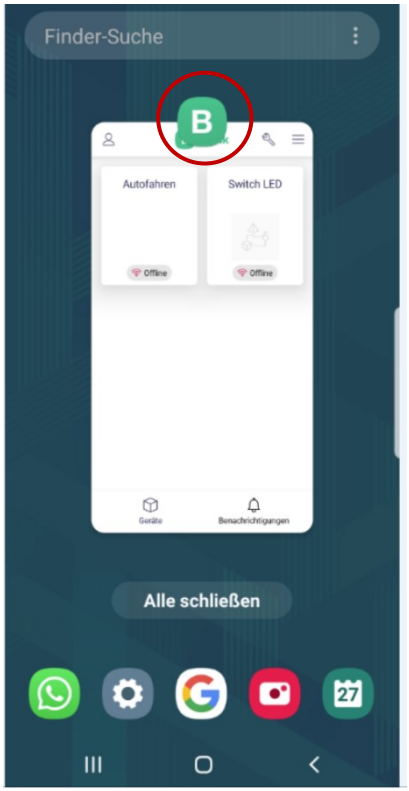
Siehe Folgeseiten.

Sketch 52 Auto fahren über Blynk – Kamera über WhatsApp

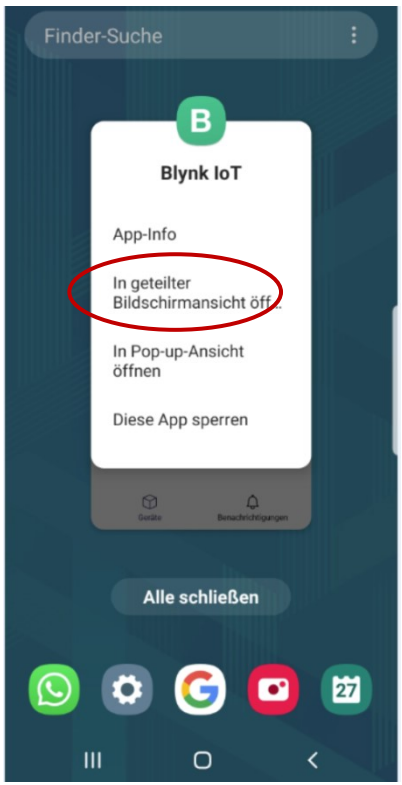
Blynk starten.
Links unten klicken.



Blynk-Icon klicken.



Geteilte Bildschirmansicht klicken.

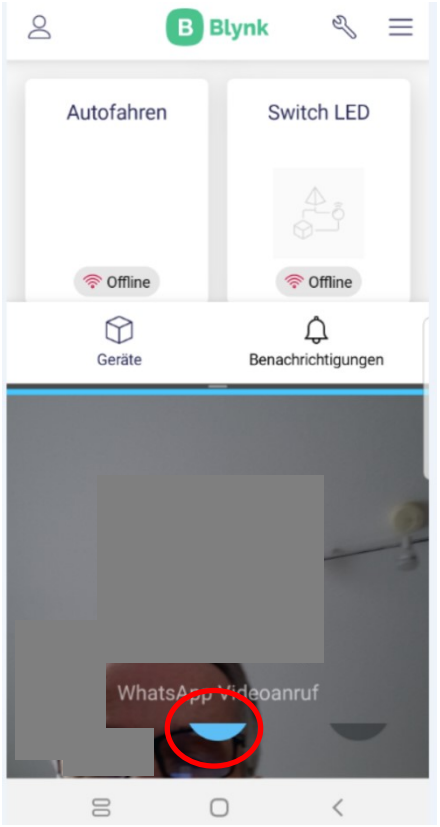


Geteilte Bildschirmansicht bereit.

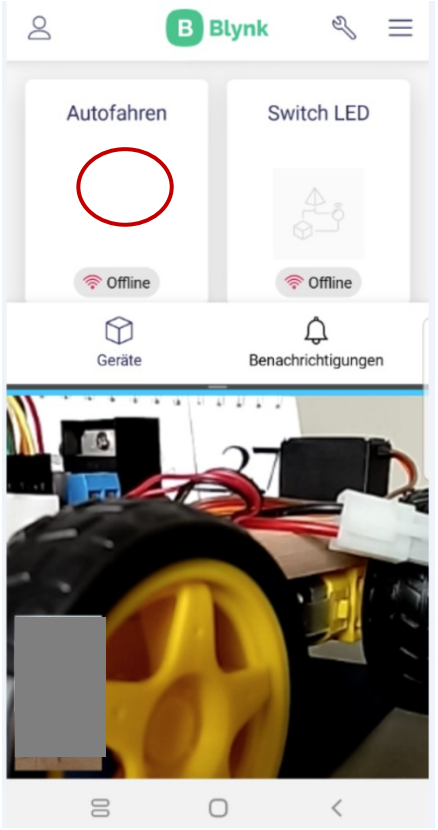
Sketch 52 Auto fahren über Blynk – Kamera über WhatsApp

Am Smartphone 2 wird der WhatsApp Videoanruf gestartet.

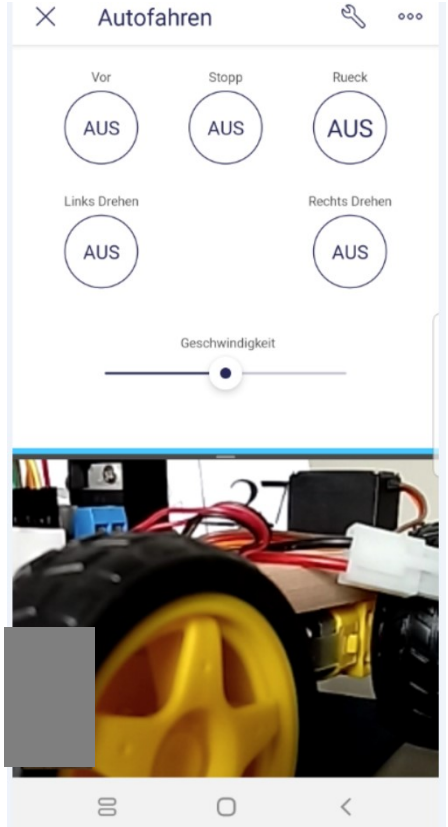
WhatsApp Anruf eingegangen und annehmen (blau hochschiebern).



Klicken „Autofahren“.



Fertig.



Bildschirm-aufteilung änderbar

ANYbotics bei BASF

ANYmal Makes the Case at BASF Chemical Plant – YouTube

Anhang

Inhaltsverzeichnis

Begriffe und Abkürzungen

AP	Access Point	Zugangs-Punkt zum WLAN (und meist auch zum Internet)
Soft AP	Access Point ohne eigenen Internetzugang	
API	Application Programming Interface	Programmierschnittstelle , zu anderen Programmen
DHCP	Dynamic Host Configuration Protocol	der Router teilt jedem WLAN-Client eine IP-Adresse zu
DNS	Domain Name System	
DNS-Server		Speichert und liefert die IP-Adressen von Domains
Domain		Name, unter dem Websites zu finden sind
Function	A group of instructions for a specific task	eine Reihe von Befehlen innerhalb einer Software
Gateway		koppeln völlig unterschiedliche (heterogene) Netze miteinander
HTML	Hypertext Markup Language	Sprache zur Erstellung von Webseiten
HTTP	Hypertext Transfer Protocol	Protokoll für die Verknüpfung der Dokumente im WWW
IoT	Internet of Things	
IP	Internet Protocol	
JS	Java Script	
JSON	JavaScript Object Notation	Eine JSON-Datei speichert Code in Form von Objekten, Arrays, ... Der Zweck ist, Daten zwischen verschiedenen Programmen oder Server-Client-Systemen auszutauschen.
MAC-Adresse	Media Access Control-Adresse	Jedes Gerät in einem Netzwerk hat eine feste Hardware-Adresse
NAT	Network Address Translation	Adress- und Portumsetzung, Verbindung zwi Heimnetz und Internet
RSSI	Received Signal Strength Indicator	Indikator für die Signalstärke des WLAN-Netzwerks an dieser Stelle
SSID	Service Set Identifier	Name eines WLAN-Netzwerks (bezeichnet den Access Point AP)
TCP- Verbindung	Transmission Control Protocol	
UDP-Verbindung	User Datagram Protocol	
URL	Uniform Resource Locator	Webadresse

Weblinks zu ES8266

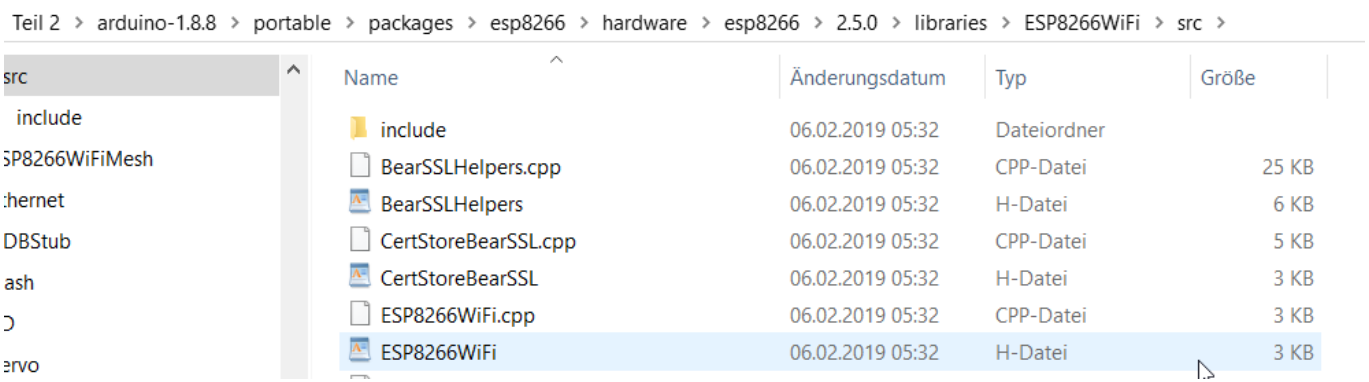
https://www.mikrocontroller-elektronik.de/nodemcu-esp8266-tutorial-wlan-board-arduino-ide/

http://stefanfrings.de/esp8266/

Wo findet man die ESP8266 Bibliotheksprogramme ?

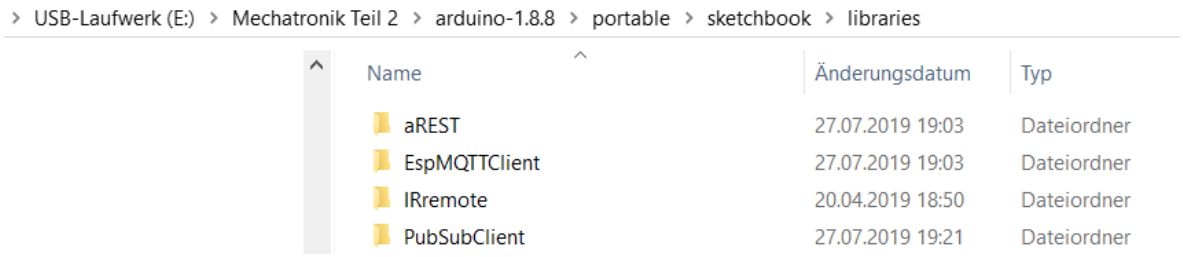
...\arduino-1.8.8\portable\packages\esp8266\hardware\esp8266\2.5.0\libraries

Zum Beispiel `<ESP8266WiFi.h>`



oder

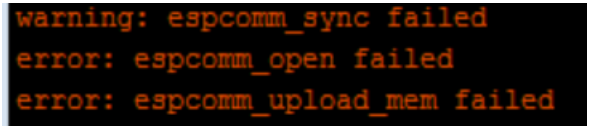
...\arduino-1.8.8\portable\sketchbook\libraries



ESP8266 … mögliche Probleme beim Upload der sketches

Wenn man den ESP Chip mit der Arduino IDE programmieren möchte, kann es manchmal Probleme geben und der ESP lässt sich nicht beschreiben. Die Fehlermeldung lautet dann oft wie folgt:

warning: espcomm_sync failed
error: espcomm_open failed
error: espcomm_upload_mem failed
error: espcomm_upload_mem failed



Um das Problem zu beheben, muss man zunächst unter „Werkzeuge“ » „Board“ prüfen, ob das richtige Board ausgewählt ist. Je nach verwendetem ESP Modul kann das zum Beispiel „NodeMCU o.9“, „NodeMCU 1.0“, „Wemos D1“ oder einfach das „Generic ESP8266 Modul“ sein.

Anschließend muss man prüfen, ob unter „Werkzeuge“ » „Port“ auch der richtige Port ausgewählt ist.

Sind die Einstellungen korrekt, kann man die Übertragungsgeschwindigkeit noch anpassen: Auch wenn sich der ESP8266 in der Regel auch mit 115200 Baud oder sogar 230400 beschreiben lässt, hilft manchmal ein Senken der Geschwindigkeit auf 57600 Bit pro Sekunde.

Manchmal funktioniert aber auch: Abziehen des USB-Kabels, mit dem das ESP Board an den Rechner angeschlossen ist. Direkt nachdem erneuten Einstecken des Kabels kann man sein Programm dann erneut kompilieren und übertragen – in der Regel funktioniert das Hochladen dann direkt ohne den „espcomm_upload_mem failed“ Error!

https://arduino-esp8266.readthedocs.io/en/latest/faq/a01-espcomm_sync-failed.html

Eine Besonderheit … der yield() – Befehl

Da sich der ESP-Chip im Hintergrund immer auch um das WLAN kümmern muss, darf der ESP-Chip durch eigene Programme nie länger als 20 ms an einem Stück blockiert werden.

Bei längeren Schleifen oder Operationen muß ein **yield()** oder **delay(…)** – Befehl eingefügt werden.

Denn yield() gibt dem Chip wieder Zeit für seine Aufgaben. Würde der ESP-Chip nicht mehr genügend Zeit bekommen, so würde er sich aufhängen, also nicht mehr reagieren.

Der yield() Befehl wird inzwischen auch in anderen Funktionen der Library, wie z.B. dem Befehl **delay** selbst schon aufgerufen.
Würde man aber eine Endlosschleife ohne delay und yield programmieren, so wäre ein Absturz sicher.

Also im Zweifel bei Schleifen die länger als 20ms dauern, ein yield oder delay einfügen.

Einschränkungen bei Bibliotheksprogrammen

Einige Bibliotheksprogramme (Libraries) funktionieren nicht mit dem NodeMCU-Board.

Das hat auch mit der yield()-Problematik zu tun.
Libraries, die für andere Boards entwickelt wurden, kannten das nicht.

Die Beendigung einer loop-Schleife mit **for (;;) {}** wie bei Arduino funktioniert beim ESP nur eingeschränkt:

Ja, bei Start des Sketches nach Upload.

Nein, wenn der Sketch mit Reset neu gestartet wurde.

Alternativ: Den gesamten Sketch im `void setup()` schreiben.

NodeMCU Modul V2 und Modul V3

Das Modul V2 hat die Abmessungen (LxBxH): 48x26x13 mm und ist geeignet für Verwendung auf einem Breadboard.

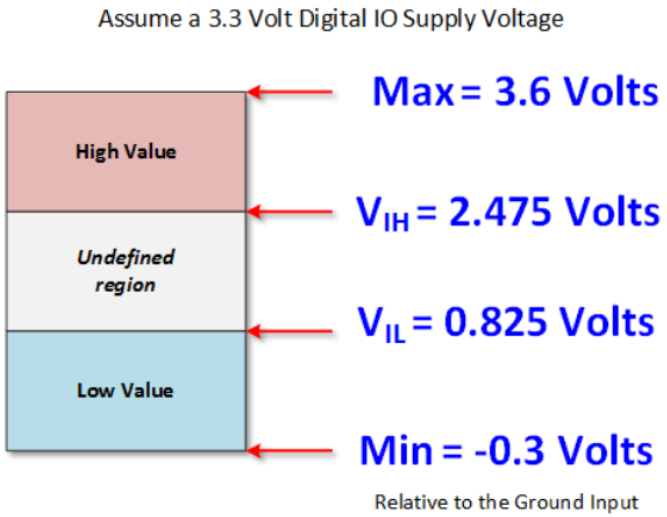
Das Modul V3 ist größer: 58x31x13 mm und ist mit 31 mm zu breit.

Weitere Unterschiede:

Bei Modul V3 sind die bisher nicht benutzten Pins RSV (Reserve) an GND und an UU (USB Power Out) angeschlossen.

Das Modul V3 hat anstatt des USB-Treibers CP2102 den Treiber CH340G (muß installiert werden, wie bei Arduino Nano im Teil 1).

High- und Low-Pegel bei 3,3V Versorgungsspannung



ESP8266 versus ESP32

Eigenschaft	ESP8266	ESP32
Mikrocontroller	single core	dual core
Max Frequenz	160 MHz	240 MHz
Flash	4 MB	4 MB
SRAM	160 KB	520 KB
GPIO	17	36
Touch-Sensoren	–	10
ADC Kanäle	1	16
ADC Auflösung	10-Bit	12-Bit
ADC Low-Noise Amplifier	–	Ja
DAC (Digital-to-Analog Converter)	–	1
WiFi	802.11 b/g/n	802.11 b/g/n
Bluetooth	–	Bluetooth classic und BLE Class-1..3 +12 dBm Senderleistung
CAN 2.0	–	1
I2C	1	2
PWM Kanäle	8	16
Temperatur-Sensor	–	on-chip
Hall-Sensor	–	on-chip
Stromverbrauch (Aktiv)	80 mA	260 mA
Betriebsspannung	2,3 bis 3,6 V	2,3 bis 3,6 V

Nachteile ESP32 vs ESP8266:

- Höherer Preis
- Höherer Stromverbrauch

Blynk - was sind virtuelle Pins (virtual pins) ?

Virtuelle Pins (z.B. „V0“) dienen zur Übertragung von Daten, sie sind faktisch die Bezeichnung für einen Datenkanal zwischen Blynk-Server und dem Gerät (ESP bzw. Arduino).

Sie sind nicht identisch mit den Input und Output-Pins auf dem ESP bzw.Arduino.

Im Sketch wird abgefragt, ob sich der Wert eines virtuellen Pins geändert hat.
Es muß eine Routine geschrieben werden, was dann ausgeführt werden soll.
Bsp.: Bei Änderung von „V0“ soll der Ausgang D0 auf dem ESP8266 (auch verbunden mit der internen LED) geschaltet werden.

To read data from Blynk app widgets

Use this block of code:

```
BLYNK_WRITE(V5) // V5 is the number of Virtual Pin
{
  int pinValue = param.asInt();
}
```

Where `param.asInt()` is the value from V5.

https://docs.blynk.io/en/blynk.edgent/api/virtual-pins

Hotspot erzeugen (Tethering)

Der Begriff kommt vom englischen Verb *to tether*, was *anleinen* oder *anbinden* bedeutet.

Mit Tethering macht man das Smartphone zum Modem und nutzt es, um mit dem PC Internet zu gehen.

Die Verbindung zwischen Smartphone und dem Computer, mit dem man ins Internet will, wird über USB oder Bluetooth hergestellt oder man richtet das Smartphone als WLAN-Hotspot ein.

Einstellungen - Verbindungen – Mobile Hotspot und Tethering

Macht man das Smartphone zum WLAN-Hotspot, wird ein öffentlicher Zugangspunkt geschaffen, den prinzipiell jeder mit einem WLAN-fähigen Gerät nutzen kann.

Wähle daher immer die WPA2 Verschlüsselung und vergebe ein starkes Passwort.

Der sicherste Weg ist und bleibt derzeit jedoch die Verbindung des Smartphones mit dem Computer über ein USB-Kabel – dort gibt es so gut wie keine Sicherheitsrisiken.

Ende

Inhaltsverzeichnis