

Quickstart: Internet der Dinge I (Web-Server V2)

Unser ESP8266 besitzt ein WLAN-Interface, mit dem wir Zugriff auf das weltweite Internet erlangen können. Hier liegt der eigentliche Mehrwert unserer IoT²-Werkstatt: Wir können Messdaten mit dem Smartphone oder Notebook wie gewohnt im Webbrowser anzeigen, oder wir speichern die Daten auf einer Cloud-Plattform und können den zeitlichen Verlauf jederzeit retrospektiv visualisieren und ggf. z.B. mit einer Tabellenkalkulation weiterverarbeiten. Mit Hilfe der Cloud-Dienste können wir uns sogar automatisch per Twitter-Nachricht über die aktuellen Messwerte informieren, oder wir lassen uns vom Dienst IFFF.com auf unserem Smartphone anrufen, wenn der aktuelle Messwert eine bestimmte Grenze überschreiten. Aber der Reihe nach. Hier bauen wir jetzt einen eigenen Web-Server.

Ich habe gar kein WLAN an meinem Aufstellungsort, kann ich trotzdem mitmachen?

Unser ESP8266 ist sehr flexibel und kann sogar ein eigenes WLAN aufspannen. Dazu nutzen wir einen AccessPoint-Block aus dem Baukasten „IoT-Internet-Zugang“, den wir im Setup einfügen. Mit diesem Access-Point können sich jetzt beliebige Smartphones oder Notebooks verbinden (vorausgesetzt, sie kennen das WPA2-verschlüsselte Passwort). Sinnvollerweise wählen wir einen aussagekräftigen Namen für das Netzwerk. Bei mehreren Klassen in der Schule z.B. AmpelKlasse9b. Aufgrund der begrenzten Ressourcen ist die Anzahl der gleichzeitig möglichen Clients auf 4 begrenzt.

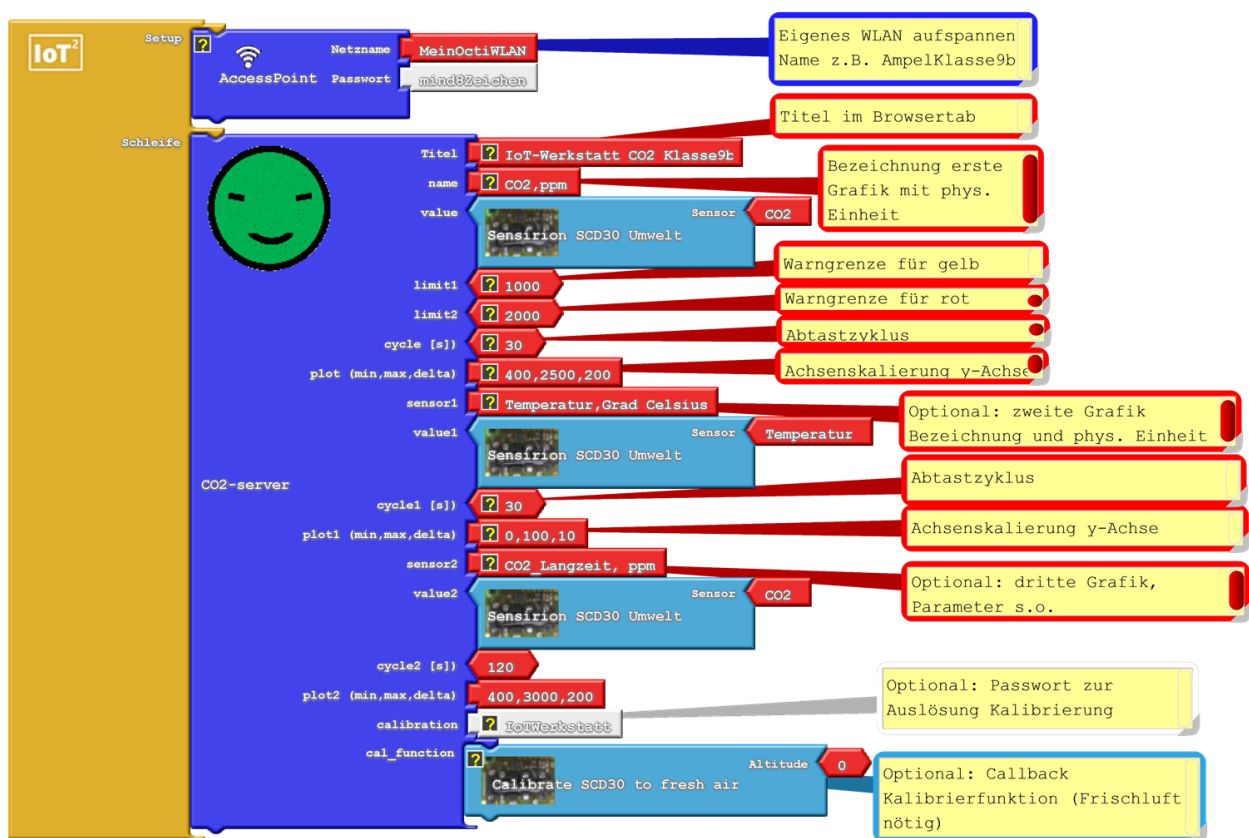


Fig. 1: Unser Mikrocontroller spannt ein eigenes WLAN auf, mit dem wir unsere anderen Geräte verbinden können. Ein Internet ist nicht nötig.

Im Baukasten „IoT:COVID19“ finden wir einen neuen, extra für die CO₂-Ampel angepassten Web-Server. Dank der Server-Bibliothek¹ sind wir in der Lage, jetzt auch eine Zeitreihendarstellung der Messwerte im Browser zu ermöglichen. Dabei gibt es maximal drei Kanäle, die unabhängig voneinander darstellbar sind. Der erste Kanal ist der Hauptkanal und bestimmt mit seinen Warngrenzen (limit1, limit2) die Farbe und das Aussehen des Smileys auf der Seite. Die anderen Kanäle sind optional und gestatten die Darstellung weiterer Messgrößen. Im Abb. 1 nutzen wir zusätzlich zum CO₂-Wert im Kanal 2 noch die Lufttemperatur im Raum. Insgesamt werden 128 Messpunkte pro Kanal gespeichert und dargestellt. Bei einem Abtastzyklus von 30 Sekunden umfasst die Darstellung mit 64 Minuten eine gute Schulstunde inklusive Pausen. Wollen wir längere Trends visualisieren, so vergrößern wir einfach den Abtastzyklus. Im Kanal 3 haben wir bei 120 s rund 4 Stunden im Blick (128*120 s = 256 Minuten). Außerdem bietet der Server die Möglichkeit eines Exports der Messdaten als csv-Datei. Diese lässt sich z.B. mit einer Tabellenkalkulation am PC sehr einfach weiterverarbeiten. Da unsere Ampel nicht über eine Echtzeituhr verfügt, fehlt in der csv-Datei der zeitliche Bezugspunkt. Dieses Manko können wir beim Speichern der Datei z.B. durch Wahl des Dateinamens (CO₂_Daten_12012021_1111.csv) ausgleichen. Das Wichtigste zum Schluss: Unseren Ampel-Server erreichen wir im Web-Browser unseres verbundenen Smartphones über die feste Adresse 192.168.4.1.

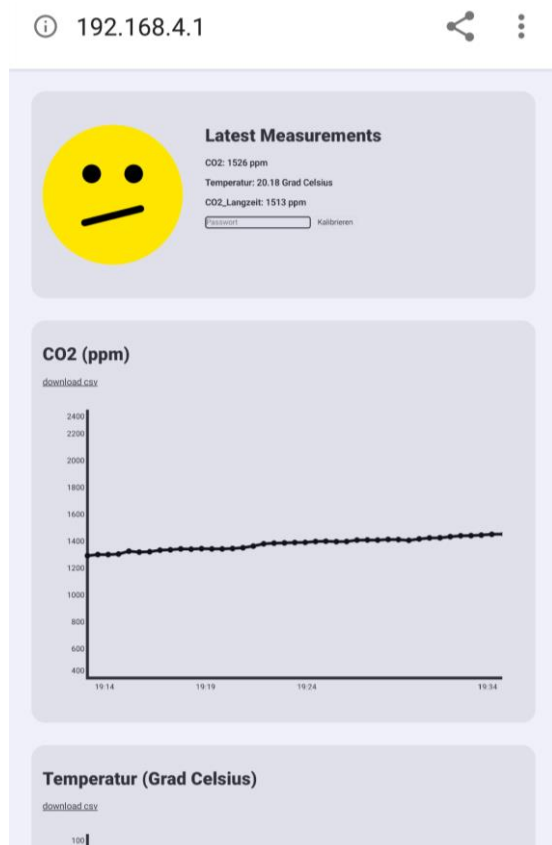


Fig. 2: Dynamische Anzeige unserer Messwerte im Web-Browser. Die Anzeige wird automatisch zyklisch aktualisiert. Je nach Grenzen ändert sich die Farbe des Smileys. Messwert kleiner limit1: grün, größer limit2: rot, dazwischen gelb.

¹ https://github.com/jason-rietzke/SensorPlot_WebInterface

Wie bereits erwähnt, besteht im CO2-Server-Block die Möglichkeit, zwei weitere Messgrößen als Tabelle anzuzeigen. Docken wir hier unsere Sensoren an, dann sehen wir die aktuellen Messdaten auf der Webseite. Bezeichnung, physikalische Einheit, Abtastzyklus und Skalierung der y-Achsen können im Anschluss „plot“ konfiguriert werden. Hier erfolgt die Angabe der Darstellungsgrenzen (min, max) und des Beschriftungsgitters (delta), jeweils durch Komma getrennt.

Zusätzlich besteht die Möglichkeit, ein Passwort für die Kalibrierung zu vergeben. Dann erscheint im Browser ein entsprechendes Eingabefeld und bei Eingabe des richtigen Codes wird die Kalibrierfunktion aufgerufen.

Wichtig: Kalibrieren nur wenn sichergestellt ist, dass der Sensor auch für eine längere Zeit frische Außenluft sieht (s. Anleitung). Kein Durchzug!

Ich habe bereits ein WLAN – wie kann ich den Octopus damit verbinden?

Ist bereits ein WLAN mit WPA2-Zugang vorhanden, so brauchen wir nur den Block für den Internetzugang auszutauschen und Netzwerkname bzw. Passwort zu konfigurieren. Nach Verbindung mit dem Access-Point bekommt der Mikrocontroller eine eigene Adresse zugewiesen. Diese Adresse können wir im Serial-Monitor anschauen und per Cut & Paste in den Web-Browser übernehmen. Ansonsten läuft alles analog. Auch hier dürfen wieder nur wenige Clients (d.h. Web-Browser) gleichzeitig auf unsere Ampel zugreifen. Auch ein vom Smartphone geteilter WLAN-Zugang (Hotspot) ist möglich. Achtung: Manche Access-Points verbieten die Kommunikation zwischen Gast-Clients (so z.B. das Fritzbox-Gastnetz). Dann muss diese Kommunikation im Admin-Bereich des jeweiligen Access-Points freigeschaltet werden. Auch ist unser Server (glücklicherweise) nicht von außen (vom Internet) aus sichtbar, sondern nur im gleichen lokalen Netz.

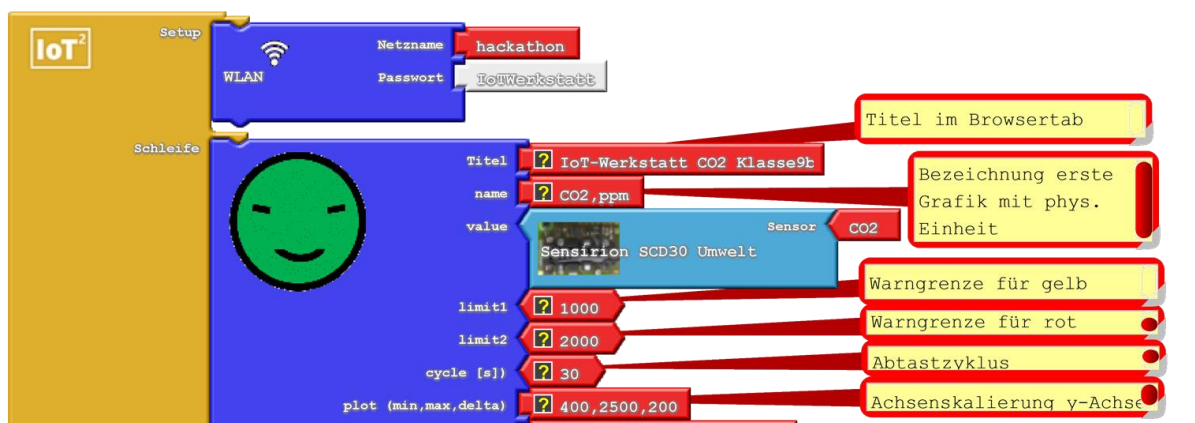


Fig. 2: Integration der CO2-Ampel in ein bestehendes WLAN. Der vorhandene Accesspoint vergibt die IP-Adresse im Netz.

```
COM34
18:40:00.167 -> .....
18:42:48.769 -> connected, meine IP:192.168.137.209
18:44:47.194 ->
18:44:47.194 -> WLAN connect to:hackathon.....
18:44:51.429 -> connected, meine IP:192.168.137.209
```

Fig. 1: Die vom Access-Point vergebene IP-Adresse wird im serial- Monitor angezeigt

Wem die Eingabe der IP-Adresse zu mühselig erscheint, der kann das Multicast Domain Name Servive-System nutzen (mDNS, im Baukasten Internet-Zugang). Ähnlich wie beim DNS im Internet sorgt hier ein lokaler Dienst für die Auflösung der IP-Adresse. Im Ardublock-Programm vergeben wir dazu einen lokalen Namen (z.B. AmpelKlasse9b), den wir im Browser dann per <http://AmpelKlasse9b.local> aufrufen können. Leider bietet Android zur Zeit keine Möglichkeit zur Nutzung dieser Komfortfunktion auf dem Handy, hier bleibt uns dann nur die Eingabe der IP-Adresse im Browser².



Fig. 3: Mit Verwendung des mDNS-Blocks im Setup ist der Server im lokalen Netz per Namen erreichbar.



IoT²-Werkstatt

² <https://ttapa.github.io/ESP8266/Chap08%20-%20mDNS.html>