

Der CO₂-Warner für die Schule

Lüften, lüften, lüften – um in Coronazeiten das Infektionsrisiko in kleinen Räumen zu begrenzen, muss regelmäßig die Luft ausgetauscht werden. Wann es Zeit ist, die Fenster aufzureißen, kann eine selbstgebaute CO₂-Ampel anzeigen.

von Guido Burger, Richard Fix und Klaus-Uwe Gollmer

Kaum ein Thema bewegt unsere Gesellschaft im Moment mehr als die COVID-19-Pandemie, besonders der Start des Regelbetriebs an Schulen und Hochschulen ist eine Herausforderung. Hier gilt es, die Ausbreitung von SARS-CoV-2, wie das Virus genannt wird, durch organisatorische Maßnahmen zu verhindern. Inzwischen ist klar, dass die Viren vor allem über Tröpfchen übertragen werden, die beim Atmen, Sprechen und Husten entstehen. Große Tropfen fallen schnell zu Boden, sodass Abstand halten hilft. Mittlere Tropfen werden durch das Gewebe von Masken an der Ausbreitung gehindert. Aber was ist mit den feinsten Tröpfchen?

Diese Aerosole sind unsichtbar, schweben lange Zeit in der Luft und werden durch Verdunstung immer kleiner, bis nur noch die festen Bestandteile übrig bleiben. In den Tröpfchen vorhandene Viren können so bis zu Stunden infektiös bleiben. Kritisch ist dies vor allem in kleinen Innenräumen mit vielen Personen. Hier hilft nur moderne Klimatechnik oder lüften, lüften, lüften. Aber auch Lüften muss gelernt sein, denn zu viel Lüftung sorgt im Winter für unnötigen Wärmeverlust und eine schlechte Klimabilanz. Darüber hinaus möchte niemand unnötig frieren.

Mitte August empfahl das Umweltbundesamt daher, in jeder Unterrichtspause intensiv zu lüften. Außerdem, so das Amt, könnten „CO₂-Ampeln als Anhaltspunkt für gute oder schlechte Lüftung dienen“. Fertige Ampeln gibt es im Preisrahmen von 80 bis 170 Euro. Wir setzen aber auf DIY-Lösungen, an deren Bau und Programmierung gleich in mehreren Schulfächern angeschlossen werden kann – von der Informatik bis zur Chemie. Unser Modell arbeitet mit dem IoT Octopus als Mikrocontroller, der vielfältige Auswertungen unterstützt, von einer bunten Neopixel-LED-Ampel über Trendanzeigen auf einer LED-Matrix bis hin zu IoT-Plattformen wie Thingspeak und Grafana.

CO₂ und VOC

Um herauszufinden, wann wir lüften müssen, nutzen wir einen biologischen Zusammenhang: Ausatemluft enthält mehr CO₂ und volatile organic compounds (VOC, auf deutsch flüchtige organische Verbindungen) als die eingeatmete Luft. Deren Konzentrationen werden üblicherweise in ppm (parts per million) gemessen. Mit ca. 400 ppm CO₂ (0,04 vol%) und in der Regel weniger als 0,5 ppm VOCs enthält die normale Frischluft nur sehr wenig dieser unsichtbaren Gase. Die Konzentration in der Ausatemluft beträgt dagegen 40.000 ppm oder 4 vol% CO₂ und damit deutlich mehr. Da sich auch die potentiell gefährlichen Aerosole in der Ausatemluft befinden, können wir deren Ausstoß nun abschätzen, indem wir die anderen Gase messtechnisch erfassen.

Kurzinfo

- » Luftqualität verstehen
- » Richtig lüften
- » CO₂-Ampel mit IoT Octopus bauen

Checkliste



Zeitaufwand:

2 Stunden



Kosten:

ab 40 Euro



Holzbearbeitung:

Schneiden mit dem Lasercutter

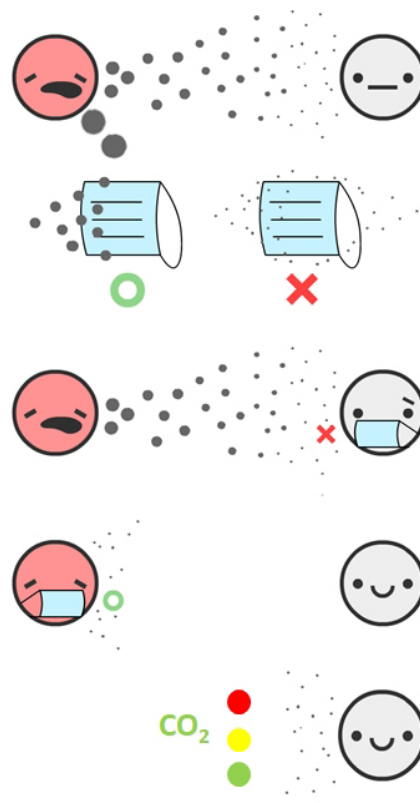
Material

- » IoT Octopus mit BME680 (oder anderes ESP8266-basiertes Board)
- » Grove CO₂-Sensor (SCD-30) mit Kabel
- » Adafruit Charlie Wing
- » Micro-USB-Kabel (5 Volt)
- » 3mm Birken-Sperrholz
- » 9x Nylon-Schrauben und Muttern (2,5mm)
- » 4x Schrauben

Mehr zum Thema

- » Guido Burger, IoT.Octopus, Make-Sonderheft IoT, S. 16
- » Florian Schäffer, Umweltsensoren für Mikrocontroller, Make 1/19, S. 8
- » Helga Hansen, Umweltsensoren einfach eingesetzt, Make 1/20, S. 16

Alles zum Artikel
im Web unter
make-magazin.de/xyrg



Große Tropfen fallen zu Boden (Abstand), mittlere Tröpfchen werden beim Wegfliegen durch Verdunstung kleiner (Aerosole).

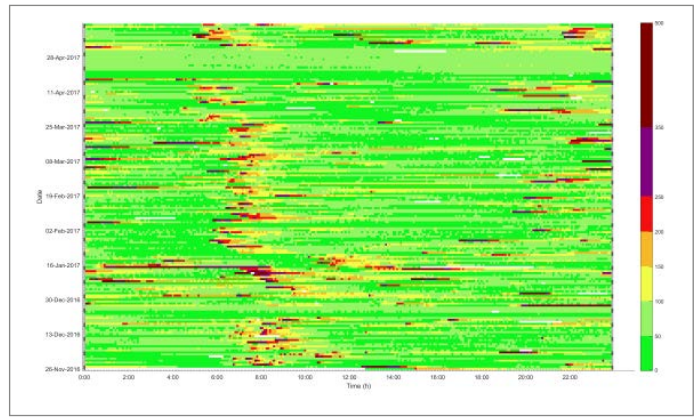
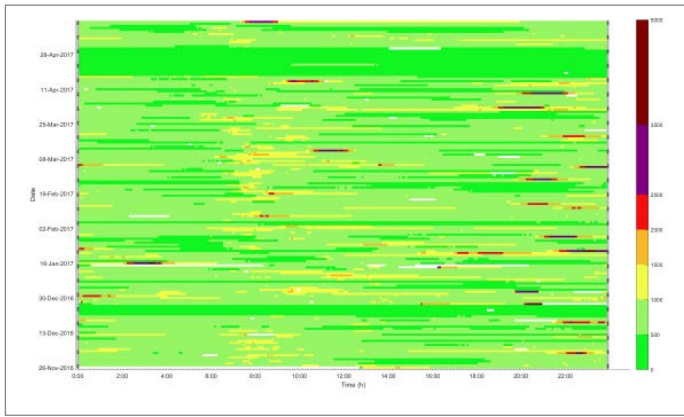
Masken halten größere Tröpfchen besser auf, kleinere schlechter.

Deshalb sind Masken weniger gut dazu geeignet, sich vor einer Erkrankung zu schützen.

Aber effektiv dabei, dir zu helfen, andere vor einer Ansteckung zu schützen.

Eine CO₂-Ampel dagegen kann dich vor möglichen Aerosolen warnen.

Adaptiert von tquev.github.io/covid-19, einer spielbaren COVID19-Simulation



Vergleich von Sensordaten in einem Badezimmer (links CO₂, rechts VOC)

Beim Bau einer Ampel, die zum Lüften auffordert, sind prinzipiell sowohl Messungen von CO₂ als auch von VOCs nutzbar. Der grundlegende Unterschied ist, dass CO₂-Messungen nur ein Gas in der menschlichen Ausatemluft ermitteln, während VOC-Sensoren fast alle Verunreinigungen der Raumluft durch chemische Stoffe erfassen. Sind allein Menschen der Grund für schlechte Luft, führt dies zu äußerst

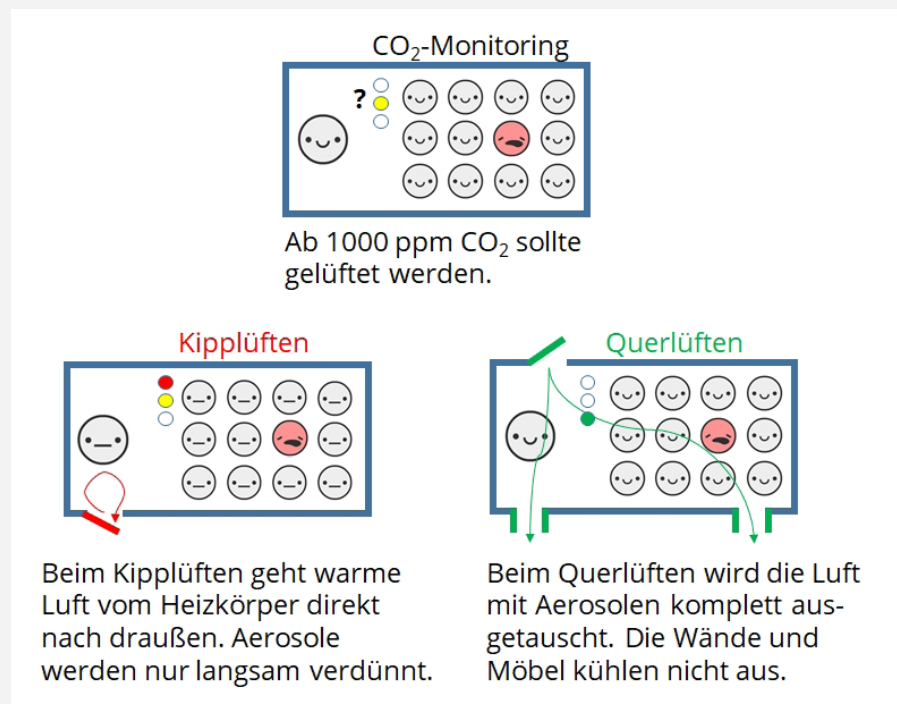
ähnlichen Ergebnissen, etwa im Schlafzimmer. Einen deutlichen Unterschied gibt es dagegen in Badezimmern, Küchen und Esszimmern oder wenn schlechte Luft von außen kommt, beispielsweise Verbrennungsgase. Auch Ausdünstungen aus Teppichen und Gegenständen oder der Einsatz von Putzmitteln wird von VOC-Sensoren erfasst. Für eine automatische Lüftungsregelung in Häusern sind sie deshalb

zu bevorzugen, weil sie die allgemeine Luftqualität erkennen. Außerdem sind sie deutlich günstiger, kleiner und bringen oft eine Temperatur-, Feuchte- und Luftdruckmessung mit, was ebenfalls der Einschätzung der Luftqualität dient. Da die meisten Schulen keine Lüftungsanlagen haben, ist hier eine Anzeige nötig, die ein Zeichen liefert, wenn das Lüften notwendig ist bzw. beendet werden kann.

Richtig lüften

Kippen oder ganz öffnen, ein oder mehr Fenster – wie lüftet man nun richtig? Hier gibt es viele Möglichkeiten und viele Fehlerquellen. Das Ziel ist der möglichst komplette Austausch der belasteten Luft durch frische Luft von draußen. Am sinnvollsten ist eine kurze Querlüftung, bei der zwei gegenüberliegende Fenster weit geöffnet werden und kurz Durchzug erlauben. Dadurch wird eine hohe Wechselrate erzielt und wir haben schnell wieder frische Luft. Ist Querlüftung nicht möglich, sollten so viele Fenster wie möglich weit geöffnet werden. Außerdem empfiehlt das Umweltbundesamt, in Räumen, in denen Sport getrieben wird, fünfmal pro Stunde auf diese Weise zu lüften und nach Auftreten von Krankheitssymptomen wie wiederholtem Husten oder Niesen.

Gerade im Winter werden die Fenster oft etwas länger auf Kipp gestellt, damit niemand frieren muss. Dabei hat Luft eine vergleichsweise geringe spezifische Wärmekapazität und speichert nur wenig Wärme. Kalte Luft wird im Raum nach kurzer Querlüftung daher sehr schnell erwärmt. Das warme Mauerwerk und die Möbel dienen quasi als große Flächenheizung. Bei Kipplüftung ist der Luftaustausch



dagegen sehr schlecht. Im schlimmsten Fall tauschen wir nur die warme Luft über der Heizung, während die aerosolhaltige Luft im Raum stehen bleibt. Da die Warm-

luft vom Heizkörper nach draußen strömt, wird es im Raum kälter und wir denken, wir hätten genug gelüftet – in Corona-Zeiten ein gefährlicher Trugschluss.

Dafür gilt die CO_2 -Konzentration bereits seit langem als guter Indikator mit klaren Grenzwerten. CO_2 -Werte sind außerdem einfach zu verstehen und durch das lineare Messprinzip der Sensoren kann man auf die Anzahl der Personen im Raum schließen, was in Corona-Zeiten ebenfalls sinnvoll sein kann.

Messprinzipien

Moderne CO_2 -Sensoren messen die Lichtabsorption von CO_2 -Molekülen im Infrarotbereich, bei $4,3 \mu\text{m}$, und werden daher auch nichtdispersive Infrarotsensoren (NDIR) genannt. Da dies eine entsprechende optische Weglänge und Lichtleistung erfordert, sind selbst miniaturisierte CO_2 -Sensoren noch einige Kubikzentimeter groß und haben eine mittlere Stromaufnahme von 20 mA. Um verlässliche Messergebnisse zu liefern, müssen sie regelmäßig kalibriert werden. Dies ist an der Außenluft meist einfach möglich.

Seit 2017 gibt es miniaturisierte, digitale VOC-Sensoren, die flüchtige organische Verunreinigungen messen. Sie sind nur wenige Kubikmillimeter groß, günstig herzustellen und haben eine mittlere Stromaufnahme von 0,09 bis 0,9 mA, je nach Betriebsmodus. Das zugrundeliegende Messprinzip ist das der halbleitenden Metalloxide, deren elektrische Leitfähigkeit sich beim Kontakt mit VOCs ändert. Die geringste Leitfähigkeit zeigt das Metalloxid in reiner Luft. Je mehr Verunreinigungen dazukommen, umso höher wird die Leitfähigkeit. Sie steigt aber nicht linear, sondern stark bei Veränderungen kleinster Gesamtkonzentrationen und langsamer bei entsprechend großen Konzentrationen. Die Sensoren linearisieren das Signal dann.

VOC-Sensoren messen außerdem kein einzelnes Gas, sondern nahezu alle reaktiven Verbindungen in der Luft in Summe. Aus dem Messwert kann somit nicht die Konzentration einzelner Moleküle abgeleitet werden. Man-

che VOC-Sensoren errechnen aus dem VOC-Signal noch CO_2 -Äquivalente ("e CO_2 "). Falls Ausatemluft die wesentliche Quelle ist, stimmen die Werte mit CO_2 -Messungen gut überein. Im Falle anderer Quellen ist das CO_2 -Äquivalent deutlich höher – gelüftet werden muss dann aber auch.

Sowohl die Messungen von CO_2 wie auch von VOCs sind von Einflüssen wie Luftfeuchte, Luftdruck und Temperatur abhängig. Einige Module bringen daher entsprechende Sensorelemente zur Kompensation gleich mit. In fertigen CO_2 -Ampeln ist übrigens nicht immer ersichtlich, welcher Sensor zum Einsatz kommt: Ist ein optischer NDIR-Sensor verbaut oder wird ein e CO_2 berechnet? Auch die Angabe, wie die Geräte zu kalibrieren sind, fehlt manchmal. In solchen Fällen ist es schwer einzuschätzen, wie zuverlässig die Ampeldata eigentlich sind.

Grenzwerte

Das Umweltbundesamt hat allgemeine Leitlinien zur "Gesundheitlichen Bewertung von Kohlendioxid in der Innenraumluft" verfasst, an denen wir uns orientieren, um die Schwellwerte für die Ampel festzulegen. Demnach ist eine CO_2 -Konzentration unter 1000 ppm hygienisch unbedenklich. Eine Konzentration zwischen 1000 und 2000 ppm stuft die Leitlinie als bedenklich und alles darüber als inakzeptabel ein. Ab 1000 ppm sollte danach mit dem Lüften begonnen werden und über 2000 ppm muss gelüftet werden. Führt dies zu Störungen des Unterrichts, sollten Maßnahmen wie eine Verringerung der Personenzahl im Raum oder der Einbau einer Lüftungsanlage überprüft werden.

Der CO_2 -Wert ist dabei nicht nur Indikator für die Verbreitung von Aerosolen, sondern auch für das allgemeine Wohlbefinden. Allerdings beeinflussen dies vor allem die chemisch reaktiven VOCs, welche mit dem menschlichen

SCD-30-Anschlüsse

SCD-30	Grove-Kabel
VIN	rot
GND	schwarz
SCL	gelb
RX/SDA	weiß

Organismus wechselwirken und somit beispielsweise zu Unwohlsein führen können. In einer weiteren Handreichung empfiehlt das Umweltbundesamt, die Summe der organischen Verunreinigungen immer unter 10 bis 25 mg/m^3 (entspricht ca. 5 bis 10 ppm, abhängig vom jeweiligen VOC) zu halten. Schließlich sollte die relative Luftfeuchtigkeit im Raum bei 40 bis 60 Prozent liegen, da Aerosole bei geringerer Luftfeuchte länger in der Luft bleiben. Diese Empfehlungen gelten aber vor allem für den Einsatz von Lüftungsanlagen.

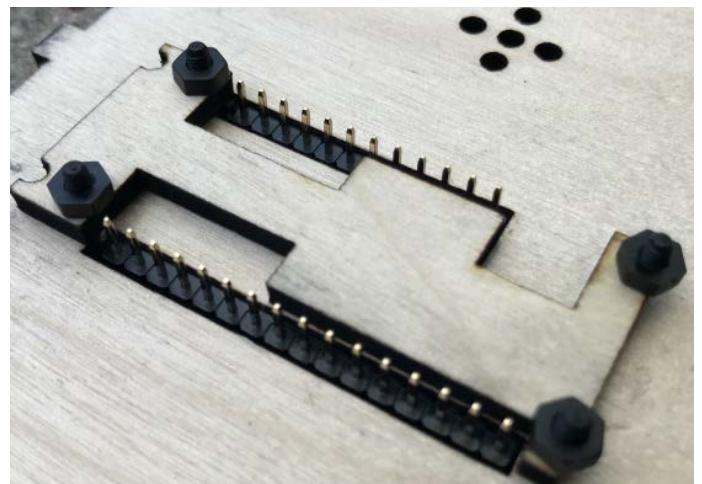
Elektronik

In unserer Ampel nutzen wir den IoT Octopus, den wir für den Einsatz im Schulunterricht konzipiert und bereits im Make IoT-Sonderheft 2016 vorgestellt haben. Die quelloffene Platine basiert auf dem WLAN-Mikrocontroller ESP8266, und verfügt neben dem VOC-Sensor BME680 und zwei Neopixel-RGB-LEDs über zwei Grove-Buchsen und einen Feather-Steckplatz, an denen weitere Hardware schnell angesteckt werden kann. Darüber hinaus haben wir den Bau mit alternativen, ESP8266-basierten Platinen, wie WeMOS D1, Adafruit Feather Huzzah oder NodeMCU getestet. Weitere Hinweise dazu finden Sie online (siehe Link in der Kurzinfo).

Mit dem integrierten VOC-Sensor BME680 lässt sich bereits näherungsweise ein CO_2 -Wert ermitteln und mit den LEDs in Ampelfarben ausgeben. Der BME misst außerdem Luftfeuchte, Temperatur und Luftdruck und kom-



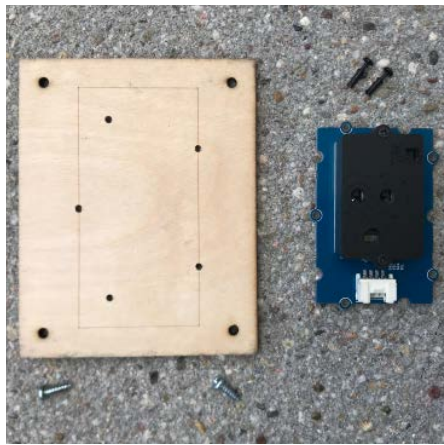
Der Sperrholzrahmen zusammengesteckt und das Display bereits montiert



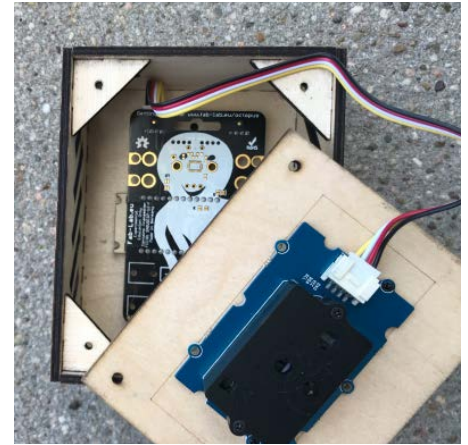
So ist das Holzstück hinter dem Display zu verschrauben.



Die Stromversorgung erfolgt über die seitliche USB-Steckbuchse.



Rückwand mit Bohrungen für die Aufnahme des CO₂-Sensors



Ein Grove-Kabel verbindet den CO₂-Sensor mit dem IoT Octopus.

pensiert so deren Einflüsse auf Messwerte. Zusätzlich verwenden wir einen NDIR-Sensor, den Sensirion SCD-30. Sein Messbereich liegt zwischen 400 und 10.000 ppm, mit einer Genauigkeit von ± 30 ppm plus 3% vom Messwert. Bei einer Messung pro zwei Sekunden benötigt er 19 mA Strom. Er besitzt eine automatische Temperaturkompensation über einen eingebauten Sensor. Die Version des chinesischen Hardwareherstellers Seeed Studio lässt sich am einfachsten in eine Ampel bauen, da diese Platine einen Grove-Anschluss mitbringt. Mit einem Grove-Kabel lässt sich der Sensor einfach mit dem Mikrocontroller verbinden. Wer eine günstigere Version kauft, muss das Kabel anlöten. Nach dem Abschneiden eines Steckers werden die vier Leitungen angeschlossen, wie in der Tabelle SCD-30-Anschlüsse angegeben.

Auf dem IoT Octopus kann optional ein Knopf als Drehregler und Button montiert werden, um den CO₂-Sensor später auf Knopfdruck zu kalibrieren. Schließlich brauchen wir noch eine Anzeige, auf der wir unsere Messwerte ausgeben. Ohne Extraaufwand geht dies mit den Neopixel-LEDs, die auf dem Board bereits verbaut sind. Sie können grün, gelb oder rot aufleuchten. Für eine detaillierte Anzeige des Werteverlaufs setzen wir eine 15x7 LED-Matrix von Adafruit ein, die auf dem IoT Octopus einfach eingesteckt werden kann. Weitere mögliche Anzeigen sind zum Beispiel

der Neopixel Featherwing von Adafruit, ein Neopixelring oder LC-Display, die alle von der verwendeten Programmierumgebung unterstützt werden.

Gehäuse

Für den dauerhaften Einsatz unserer CO₂-Ampel benötigen wir eine Verkleidung, welche die sensible Elektronik schützt und das Aufstellen erleichtert. Dazu verwenden wir mit einem Laser geschnittene 3mm-Sperrholzplatten. Die Vorlagen für verschiedene Ampeln gibt es kostenlos zum Download (siehe Kurmlink). Die Teile sind mit Verzahnungen gestaltet, so dass man sie einfach zusammenstecken kann. Dabei beginnen wir mit den Seitenteilen und den rechteckigen Halterungen der Rückseite. Das linke Seitenteil hat neben vier Lüftungsschlitzen noch eine Aussparung für das USB-Kabel, je nach Variante ist im oberen Seitenteil oder der Front ein Loch für den optionalen Schalter. Ein Tropfen Holzleim in den Ecken versteift die Konstruktion und sorgt so für bessere Haltbarkeit. Die Variante mit dem Adafruit Feather Huzzah passt übrigens genauso in dieses Gehäuse.

Das Display wird auf der Vorderseite mit einer kleinen Hilfskonstruktion verschraubt. Dazu sind die vier 2,5mm-Plastikschrauben und Muttern nötig. Die richtige Lage des Boards und des kleinen Holzstücks (siehe Bilder) ist wichtig, damit der IoT Octopus gleich

darauf passt. Stehen die Schraubengewinde zu weit über, sollten sie gekürzt werden.

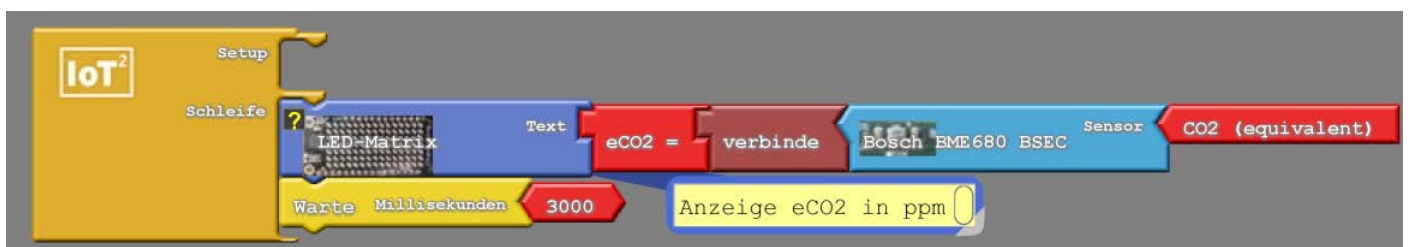
Auf die Steckerleisten kann nun der IoT Octopus aufgesteckt werden, wobei ebenfalls auf die Orientierung zu achten ist: Der BME680-Sensor gehört hinter die fünf Löcher der Frontseite. Nach dem Aufstecken der Front sollte der USB-Anschluss hinter der Aussparung auf der linken Seite zu sehen sein.

Im letzten Schritt montieren wir den Sensor. Hierzu sind in der Rückwand des Gehäuses Bohrungen vorgesehen, in denen der SCD-30 in der empfohlenen Grove-Version sehr leicht verankert werden kann. Hierzu gibt es kleine Plastikdübel von Grove. Alternativ kann man wieder 2,5mm-Schrauben nehmen, wofür die Bohrungen auf der Platine eventuell erweitert werden müssen. Ein Stück Draht oder doppel-seitiges Klebeband erfüllen die Aufgabe letztlich ebenso.

Das Grove-Kabel muss in die rechte Buchse des IoT Octopus gesteckt werden und verbindet so den CO₂-Sensor mit dem IoT Octopus. Hierbei wird der I²C-Datenbus verwendet. Mit vier Schrauben wird die Rückwand nun verschlossen.

Die Steuerung

Für die Programmierung nutzen wir IoT-Ardu-block, unsere Variante der bekannten Programmierumgebung für den Arduino. Sie



Unser erstes Programm

funktioniert mit Blöcken und umfasst spezielle Erweiterungen zum Thema Internet der Dinge (IoT). Die aktuelle Version für Windows und MacOS können Sie kostenlos auf unserer Seite herunterladen (siehe Kurzinfo). Dort finden Sie weiter eine Anleitung zur Inbetriebnahme und zur Installation des gegebenenfalls nötigen USB-Treibers, wobei es unter Windows schneller als unter MacOS geht. Gestartet wird zunächst die Datei *IoT-Werkstatt.bat*. Falls der Windows Defender dies verhindert, geht es mit Klick auf *Weitere Informationen* und *Trotzdem Ausführen* weiter. Außerdem muss der Zugriff auf „Java(TM) Platform SE binary“ zugelassen werden. Aus der Arduino-IDE heraus geht es über *Werkzeuge* schließlich zu *Ardublock*.

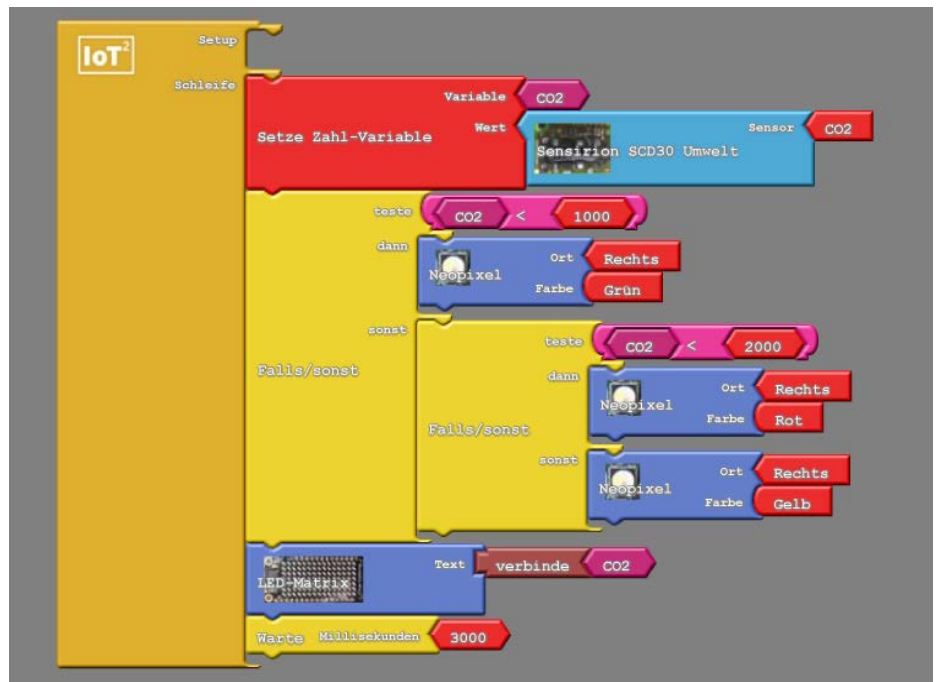
Die Ardublock-Programme bestehen, wie die übrigen Arduino-Programme, grundsätzlich aus zwei Teilen. Im *Setup* werden Anweisungen eingetragen, die der IoT Octopus nur einmal, nach dem Start, zu erledigen hat. Alle Aufgaben in der *Schleife* laufen dagegen solange wieder und wieder ab, wie der IoT Octopus angeschaltet ist. Die möglichen Anweisungen finden wir als bunte Funktionsblöcke in der linken Seitenleiste der IoT-Werkstatt. Nach dem Schlüssel-Schloss-Prinzip sind die Übergänge zwischen den Blöcken in verschiedenen Formen gehalten, sodass nur die passenden Datentypen angeschlossen werden können.

In unserem ersten Programm testen wir die LED-Matrix, den Charlie-Wing, und geben den eCO_2 -Wert des eingebauten BME680-Sensors aus. Im Setup sind keine Angaben nötig. In der Schleife setzen wir zunächst den blauen Block für die LED-Matrix (aus *Anzeigen/Aktoren*) mit dem Text *eCO=*. Der Baustein *verbinde* (aus *Kommunikation (M2M)*) sorgt dafür, dass der Sensor-Messwert in den passenden Datentyp umgewandelt wird, weshalb man beim Auswählen auf die passenden Formen achten muss. Daran wird der Sensor (*BOSCH BME680 BSEC*) angeschlossen und in seinem roten Block als auszugebender Wert *CO2 (equivalent)* ausgewählt. Nach jeder Messung soll der Octopus eine Pause von 3000 Millisekunden machen, den passenden Baustein finden wir unter *Kontrollstrukturen*.

Mit dem Einsetzen der Bausteine für die LED-Matrix und den Sensor bindet die Programmierumgebung automatisch die Treiber und Bibliotheken ein, mit denen die Bauteile angesprochen werden. Jetzt müssen wir den IoT Octopus nur noch mit einem USB-Kabel an den Rechner anschließen und das Programm mit einem Klick auf *Hochladen auf den Octopus* auf den Mikrocontroller laden.

Ausgabe als Ampel

Jetzt wollen wir Ampelfarben sehen. Dazu verwenden wir den externen CO_2 -Sensor SCD-30 und lassen die auf dem Octopus vorhandenen



Zwei geschachtelte Fallunterscheidungen – und fertig ist die Ampel

Neopixel in rot, gelb oder grün aufleuchten. Der Messwert wird wieder auf der Matrix angezeigt. Auch dieses Programm hat kein Setup. Wir beginnen mit *Setze Zahl-Variable* und nennen die Variable *CO2*. Um den vorgegebenen Wert 0 gegen den Block *Sensirion SCD30 Umwelt* (aus *Externe Interfaces*) austauschen zu können, muss man die 0 noch anklicken. Aus den *Kontrollstrukturen* holen wir dann eine *Falls/sonst-Unterscheidung* und setzen unter *teste* einen Vergleich mit *<* (aus *Log. Operatoren*). Unsere Variable *CO2* können wir mit einem Rechtsklick klonen und links im Vergleich einfügen. Rechts setzen wir die Zahl 1 (aus *Variablen/Konstanten*) und verändern ihren Wert auf 1000 – den Schwellenwert des Umweltbundesamtes. Ist dies der Fall, leuchtet über *dann* der rechte Neopixel (aus *Anzeigen/Aktoren*) grün auf. Unter *sonst* schachteln wir eine zweite *Falls/sonst-Unterscheidung* und vergleichen nun gegen die zweiten Schwellenwert von 2000 ppm. Mit der Klonen-Funktion ist dies schnell gebaut. Schließlich ergänzen wir den *LED-Matrix-Block* und lassen den Wert der Variable *CO2* über den passenden *verbinde* einlaufen. Den Abschluss macht eine Pause über 3000 Millisekunden.

Trend-Anzeige

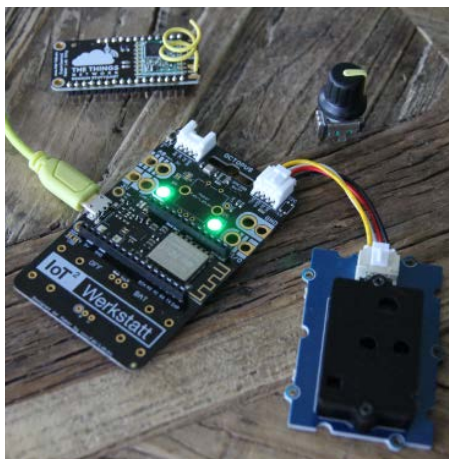
Auf der LED-Matrix können die Sensordaten über die Zeit sogar mit kleinen Balkendiagrammen dargestellt werden. So können wir einen einfachen Trend, zum Beispiel über die letzte Stunde, ablesen. Hierzu dient der Block *Schreibe Array* (aus *Variablen/Konstanten*). Das Array umfasst 15 Stellen, von 0 bis 14, der *index*, der jeweils mit einem *wert* gefüllt wird. Der

untere Block zeigt, dass diese Werte wieder vom CO_2 -Sensor kommen und befüllt den *index* mit 14. Damit die Daten mit der Zeit nach links über das Display wandern, beginnen wir das Programm insgesamt mit einer *Zählschleife* (aus *Kontrollstrukturen*). Unsere Variable nennen wir *i* und bei *Mal* setzen wir 14 ein. Hier schiebt *Schreibe Array* nun den Wert von Stelle 14 auf Stelle 13. Der Block *Lese Array* ist unter *Variablen/Konstanten* zu finden und auch die Klon-Funktion ist wieder hilfreich. Damit die Daten des Arrays auf der LED-Matrix erscheinen, fehlt nur noch *Anzeige Array* (aus *Anzeigen/Aktoren*) und eine Pause von fünf Minuten (300.000 Millisekunden). Mit den Angaben bei *min* und *max* begrenzen wir die angezeigten Konzentrationen auf Werte zwischen 380 und 1200 ppm, sodass jeder Pixel 117,14 ppm entspricht.

Nun ist unsere Ampel bestens gerüstet für den Einsatz im Klassenraum, bei dem sich die Neopixel und die Anzeigematrix quasi ständig im Blick des Publikums befinden.

Anbindung an Thingspeak

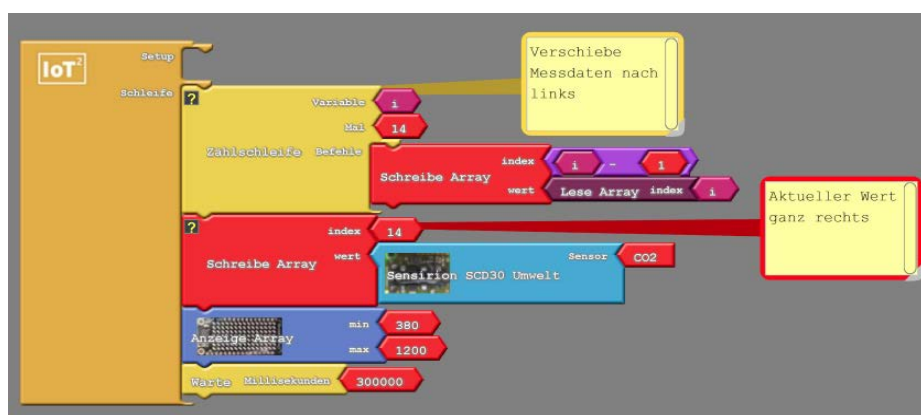
Wie sieht das Tagesprofil aus, welche maximalen Werte haben wir heute gesehen, wie dokumentieren wir den Lüftungserfolg? All diese Punkte lassen sich mittels IoT-Funktionalität quasi auf einmal klären. Dazu verbinden wir uns mit dem (hoffentlich) vorhandenen WLAN und speichern die Daten in der Cloud. Hier spielt unsere IoT-Werkstatt ihre größten Stärken aus. Ob MQTT, InfluxDB, IFTTT, oder Blynk – für alle großen Maker-Plattformen und -Dienste existieren grafische Blöckchen zur einfachen Integration.



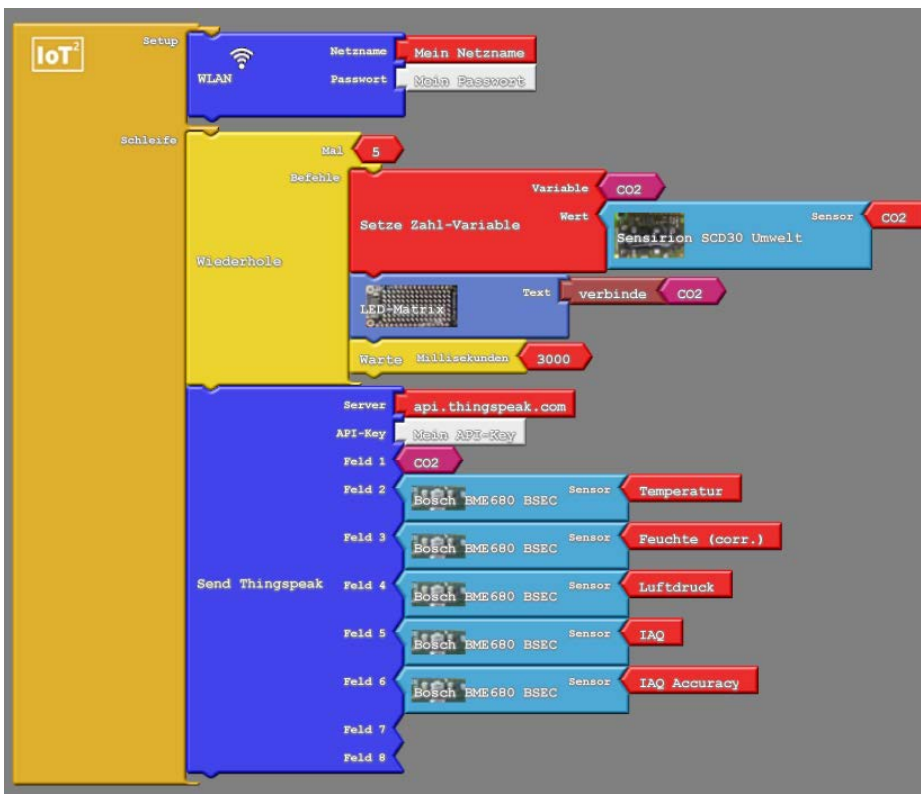
Die einfachste Ampel: Platine mit angeschloss-
nem CO₂-Sensor



Die Trendanzeige in Aktion



Programm zum Erstellen der Trendanzeige



Zwei blaue IoT-Superblöcke und fertig ist der internetbasierte Datenlogger.

Das folgende Programm nutzt die IoT-Plattform Thingspeak. Die Zugangsdaten zum WLAN tragen wir über den Block aus IoT: *Internet-Zugang* ein. Da Thingspeak in der kostenlosen Variante nur ein Update-Intervall von 15 Sekunden zulässt, aktualisieren wir mit *Wiederhole* erst vier Mal allein die Anzeige der LED-Matrix, bevor wir die verschiedenen Messwerte mit *Send Thingspeak* in die Cloud schicken. Der API-Key sorgt für die Verbindung zum richtigen Thingspeak Channel. Sie bekommen ihn nach der Anmeldung auf der Plattform. Im Make IoT-Sonderheft finden Sie auf S. 77 eine Anleitung dazu. Die vielfältigen Export- und Visualisierungsmöglichkeiten von Thingspeak ermöglichen den schul- und auf Wunsch weltweiten Zugriff auf Zeitreihen. Die Thingspeak-Serversoftware gibt es auch als Open-Source-Version zur Selbstinstallation auf einem lokalen Raspberry Pi oder anderen Rechner. Weitere IoT-Beispiele finden Sie online (siehe Kurzinfo).

Kalibrierung

Die NDIR-Sensoren kommen mit einer werkseitigen Kalibrierung und können direkt eingesetzt werden. Einmal im Jahr oder wenn die Messergebnisse unpassend erscheinen, ist eine Kalibrierung notwendig, um die vom Hersteller angegebene Genauigkeit aufrecht zu erhalten. Die typische Abweichung pro Jahr liegt beim SCD30 bei 30 ppm und maximal 80 ppm. Um die Messergebnisse zu überprüfen, kann man den Sensor für einige Stunden die frische Luft messen lassen, zum Beispiel an einem offenen Fenster. Hier sollte ein Wert um 400 ppm (± 50 ppm) vorliegen, dem normalen CO₂-Anteil. Die Außenluft dient auch als Kalibriergas. Dabei gibt es zwei Möglichkeiten:

1. Automatische Kalibrierung: In diesem Modus muss der Sensor sieben Tage laufen und in dieser Zeit jeweils für eine bestimmte Zeit an der "frischen" Luft sein. Beim SCD-30 ist es täglich eine Stunde. Der Sensor sucht dann selbständig seine Basislinie und speichert diese auf dem Sensor.
2. Erzwungene Kalibrierung: Hierbei wird der Sensor einer definierten CO₂-Konzentration (wie 400 ppm) ausgesetzt und der Basiswert angepasst. Der SCD-30 muss der Konzentration mindestens 2 Minuten ausgesetzt werden, auf jeden Fall solange, bis er einen stabilen Messwert anzeigt. In der IoT Werkstatt haben wir dafür einen Block vorbereitet: *Calibrate SCD30 to fresh air* (zu finden unter *Externe Interfaces*).

Um den Sensor mit einem Knopfdruck zu kalibrieren, muss nur eine *Falls*-Abfrage (aus *Kontrollstrukturen*) in das eigene Programm eingebaut werden. In unserer Komfort-Ausführung leuchtet nach dem Drücken des

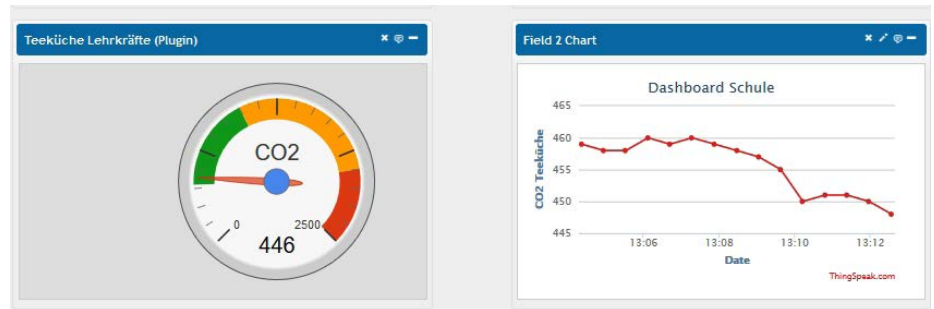
Buttons (aus Sensoren) erst die rechte Neopixel-LED blau auf und nach einer Pause von einer Sekunde beginnt die Kalibrierung.

Wer es ganz genau haben möchte, muss bei der Messung auch den Luftdruck berücksichtigen. Dafür kann man im Kalibrierungsblock die Höhe über Meereshöhe (*Altitude*) eintragen.

Einsatz (nicht nur) in der Schule

In Schulklassen bietet der Nachbau einer CO₂-Ampel Anknüpfungspunkte in vielen Fächern, über die Erläuterung der Programmierung im Informatik-Unterricht hinaus. So kann man im Physik-Unterricht klären, wie lange Tröpfchen im Raum schweben und warum die Luft beim Lüften auskühlen kann, die Wände dies aber nicht tun sollten. Die Zeitreihendarstellung auf Cloud-Plattformen macht die Auswertung und den Vergleich verschiedener Sensoren beziehungsweise Räume zum Kinderspiel und lädt zum Experimentieren ein. Wie unterscheidet sich Kipplüften vom Querlüften? Verändert die Anwendung eines Luftreinigers die Messwerte? Selbst die Kunst kann man mit ins Boot holen und kreative Alternativen für die Darstellung der Messergebnisse entwerfen – etwa auf einem Neopixelring anstelle der LED-Matrix. Weitere Ideen für den Unterricht sammeln wir auf unserer Projekt-Webseite, zu erreichen über den Link in der Kurzinfo. Schließlich sind Gemeinschaftsprojekte mit engagierten Eltern, Sponsoren aus der lokalen Wirtschaft und dem Schulträger denkbar.

Für alle, die in anderen Innenräumen arbeiten, bietet sich die CO₂-Ampel natürlich auch an. Gerade im Home Office vergisst man gerne das Lüften, dabei ist es nach langen Telefonaten dringend nötig. Und selbst Funktionsräume wie Labore, Teeküchen und Toiletten, die von wechselnden Nutzern frequentiert wer-



Die Thingspeak-Plattform visualisiert Zeitreihen – ein perfektes Tool für das Schul-Dashboard



Mit diesem Programm wird die Ampel auf Knopfdruck kalibriert.

den, können mit einer Ampel überwacht werden, um das Infektionsrisiko zu senken und trotzdem die Klimabilanz zu schonen.

Dabei kann die Ampel beim Nachbau individuell angepasst werden. Sinnvoll wäre zum Beispiel die Kombination mit einem Pax-Counter (Make 1/19, S. 32) oder zusammen mit einer lokalen Cloud. Über IFTTT können wir VoIP-Nachrichten versenden, wenn die Raumtemperatur zu niedrig und das Fenster zu lange

auf ist oder die CO₂-Konzentration zu hoch und damit das Infektionsrisiko. Code-Beispiele und Bauteillisten für Varianten mit Neopixelring oder 7-Segment-Anzeige haben wir online bereits vorbereitet (siehe Kurzinfo). Einen besonders schönen Nachbau hat Netzbastler Moritz Metz vor kurzem im Deutschlandfunk vorgestellt: Er hat unser Projekt als Inspiration für den CO₂narienvogel genommen, der bei schlechter Luft von der Stange kippt. —hch