

Vernetzte CO₂- Sensoren für Schulen



Jasper Sickinger

1. Kurzfassung

Eine Harvard-Studie¹ belegt, dass die Konzentrationsfähigkeit von Menschen bei erhöhter CO₂-Konzentration in der Umgebung stark absinkt. Bei 400 PPM über dem Außenluftniveau verliert ein Mensch nach dieser Studie rund 21% seiner kognitiven Leistungsfähigkeit. Gerade in Klassenräumen können bei normalem Atmen der Schüler CO₂-Konzentrationen auftreten, die die Leistungsfähigkeit des Gehirns stark verringern. CO₂ ist von Menschen nicht wahrzunehmen. Sie müssen also mithilfe von Geräten alarmiert werden. Bis heute waren solche Messgeräte für Schulen zu teuer und zu unflexibel. Messdaten konnten nicht zentral ausgewertet werden und die Alarmsysteme auch nicht an die Bedürfnisse der Schulen angepasst werden.

Meine rund 30 Euro teuren Sensoren ändern dies.

Mit einem WLAN-fähigen Arduino und einem günstigen NDIR-CO₂-Sensor können Schüler und Lehrer per Blinklicht über eine zu hohe CO₂-Konzentration benachrichtigt werden. Die Auswertung der gesammelten Daten ist cloudbasiert. Zudem besteht die Möglichkeit, die Sensoren in Schulprojekten nachzubauen. Auch die Effektivität der Rückmeldung konnte ich anhand eines Experiments nachweisen und damit zeigen, dass die Sensoren ihren Zweck voll und ganz erfüllen. Mein Projekt behandelt hierbei die Entwicklung der Hardware und Software der Sensoren, die Vorbereitung und Durchführung eines Schulprojekts zur Validierung der Nachbaufähigkeit der Sensoren und die Frage, inwiefern die Sensoren Menschen dazu animieren, zu lüften.

¹ Quellenverzeichnis: 2. Eintrag

2. Inhaltsverzeichnis

1. Kurzfassung.....	1
2. Inhaltsverzeichnis	2
3. Einleitung und Motivation.....	3
4. Vorgehensweise, Materialien und Methoden	3
4.1. Hardwareentwicklung	4
4.2. Software	6
4.3. Nachbau der Sensoren in einem selbstgeleiteten Schulprojekt	8
4.4. Sammeln von Daten	9
4.5. Materialliste	10
5. Ergebnisse.....	10
5.1 Gesammelte Daten zur Validierung der Nützlichkeit der Sensoren.....	11
5.1.1 Rückmeldungen der Endnutzer	13
5.2. Erfahrungswerte vom Nachbau der Sensoren im Schulprojekt:	13
5.3 Erfahrungswerte bei der Entwicklung der Hardware und Software	14
6. Ergebnisdiskussion	14
6.1. Erfahrungen aus dem Nachbau der Sensoren.....	14
6.2. Mess- und Erfahrungswerte	14
7. Fazit/Zusammenfassung:.....	15
8. Quellenverzeichnis	17
9. Unterstützungsleistungen	17

3. Einleitung und Motivation

Anfang 2019 hatte ich auf YouTube ein Video² gesehen, in dem die Einflüsse von CO₂ auf den menschlichen Körper erklärt wurden. Dabei wurde mir bei der darauffolgenden Recherche klar, dass CO₂ die kognitive Leistungsfähigkeit von Menschen sehr stark beeinflusst. In einer Studie³ wurde festgestellt, dass die Konzentrationsfähigkeit der 24 Proband*innen bei höherer CO₂-Konzentration stark sank. In der Außenluft beträgt die CO₂-Konzentration 400 ppm (parts per million). Das sind 400 Volumenanteile CO₂ pro Million Luftanteile. Bei 800 ppm wurde schon ein 21%er Verlust der kognitiven Leistungsfähigkeit festgestellt. Und für jede weiteren 400 ppm sank der Wert um weitere 21% ab. Des Weiteren gab eine Bekanntmachung des Umweltbundesamtes⁴ an, dass solche Werte in Innenräumen, vor allem in Schulen und Bürogebäuden, durchaus erreicht werden können. Dies führte zu der Idee, die CO₂-Konzentration auch in der eigenen Schule zu messen. Denn es ist bekanntlich schlecht, in einer Schule ungünstige Lernbedingungen zu haben - vor allem, wenn sie sich leicht beheben lassen. Jedoch stellte sich bald heraus, dass Fertiglösungen zu unflexibel und zu teuer waren. Eine benachbarte Schule stellte eine kommerzielle Lösung schon nach kurzer Zeit wieder ein, da sie die Schülerinnen und Schüler vom Unterricht ablenkte. Diese gaben nur den Wert aus und piepten, wenn ein fester Wert überschritten war. Außerdem gab es keine Möglichkeit, die Daten zentral auszuwerten. Insofern recherchierte ich nach der Möglichkeit, einen Sensor mit integriertem Feedback selbst zu bauen. Da ich schon einige Erfahrungen mit der Arduino-Plattform und dem dazugehörigen Microcontroller ESP8266 gemacht hatte, suchte ich zuallererst nach einer Möglichkeit, diesen zur Messung der CO₂-Konzentration zu nutzen. Dabei stellte sich heraus, dass es ein Modul namens MHZ19 gibt, das sich an einen Arduino anschließen lässt. Das Modul kostet rund 20 Euro und ist damit preisgünstig genug, um in Schulen angeschafft werden zu können. Mit diesem Modul führte ich einige Tests durch, die darin bestanden, den CO₂-Konzentration während des Unterrichts zu messen. Dies funktionierte zufriedenstellend und es wurden auch Werte erreicht, die laut der Harvard-Studie als einschränkend für die Konzentration galten. Insofern entschied ich mich, diese Sensoren weiterzuentwickeln, um sie später in einem Schulprojekt nachbauen zu können. Dieses Projekt führte ich im Juni 2019 durch.

4. Vorgehensweise, Materialien und Methoden

Meine Vorgehensweise ist in vier Teile geteilt. Diese beinhalten 1. Entwicklung und Bau der Hardware, 2. Das Programmieren der Software, 3. den Nachbau der Sensoren in einem Schulprojekt und 4. das Sammeln von Sensordaten über einen Zeitraum von rund zweieinhalb Monaten. Es gibt mehrere Fragen, die diese Arbeit bestimmen:

- Wie lässt sich ein möglichst günstiger, einfach nachzubauender Sensor verwirklichen, der auch von Schulen verwendet und nachgebaut werden kann?
- Wie muss Feedback gestaltet sein, sodass Menschen auf einen Messwert angemessen reagieren? Wie muss die Interaktion zwischen Mensch und Maschine gestaltet sein?
- Wie kommuniziert man am besten, was ein Gerät tut? Interaktion zwischen Mensch und Entwickler.
- Sind solche Sensoren als Mittel zur Luftverbesserung überhaupt effektiv?

² Quellenverzeichnis: 6. Eintrag

³ Quellenverzeichnis: 2. Eintrag

⁴ Quellenverzeichnis: 3. Eintrag

4.1. Hardwareentwicklung

Da ich mir zum Ziel gesetzt hatte, einen möglichst günstigen CO₂-Sensor zu bauen, der außerdem noch einfach anzupassen ist, war mir klar, dass ich die Arduino Plattform verwenden wollte. Mit dieser hatte ich schon gute Erfahrungen gemacht und es gibt eine gute Dokumentation, die es auch anderen Menschen erlaubt, den Code zu verstehen und zu verändern. Da die Daten der CO₂-Sensoren jedoch auch ausgewertet werden mussten, bot sich ein Controller an, der einen WLAN-Zugang hat. Die Auswertung ist zur Validierung der Nützlichkeit der Sensoren natürlich wesentlich. Aus eigener Erfahrung hatte ich den ESP8266 Microcontroller im Blick, der dem Benutzer erlaubt, sich an ein WLAN anzumelden und Daten per HTTP zu versenden.

Ein weiteres Teilziel war es auch, es Schülerinnen und Schülern möglich zu machen, die Sensoren selbst zu bauen. Dies hat den positiven Nebeneffekt, dass mit der Bekämpfung des Problems CO₂, Schülerinnen und Schülern auch das Thema Informatik und Technik nähergebracht wird. Dies brachte die Voraussetzung mit sich, beim Bau keine Werkzeuge zu verwenden, die nicht in einer Schule vorhanden sind.

Dabei orientierte ich mich an meiner eigenen Schule, die aufgrund ihrer altsprachlichen Orientierung einen guten Repräsentanten für die Mindestausstattung darstellt. Das heißt, dass es davon auszugehen ist, dass Werkzeuge, die dort vorhanden sind, auch in den meisten anderen Schulen existieren.

Bei der Recherche nach dem CO₂-Sensor Modul selbst stieß ich bald auf den MHZ19 Sensor. Dies ist ein nichtdispersiver Infrarotsensor. Dieser funktioniert nach dem Prinzip, dass CO₂ eine bestimmte Wellenlänge (ungefähr 4300nm) infraroten Lichtes absorbiert. Das heißt, dass sobald mehr CO₂ in der Luft ist, weniger Lichtintensität bei dem im CO₂-Sensor eingebauten Infrarotempfänger ankommt. Dieser Effekt variiert auch nach Temperatur. Daher ist im Sensor auch noch ein Thermometer verbaut, was auch ausgelesen werden kann und die Nützlichkeit der CO₂-Sensoren weiter verstärkt. Hierbei kann auch der Effekt des stärkeren Lüftens auf die Temperatur im Innenraum gemessen werden. Außerdem muss der Sensor am Anfang kalibriert werden. Dazu muss man ihn für ein paar Minuten an die frische Luft legen und dann zwei Kontakte miteinander verbinden. Dann wird die Außenluft mit 400 ppm als Referenzwert gespeichert. Die Differenz zu den realen 410ppm ist nicht tragisch für unseren Anwendungszweck. Die Wahl fiel vor allem auf dieses Modell, da es gegenüber chemischen Modellen, wie dem MG811, keine lange Aufwärmzeit hat und vor allem nur auf das Gas CO₂ reagiert. Das MG811 ist ein Modul, das eine Heizspannung braucht und zum Aufheizen rund 24 Stunden benötigt. Daher ist es nicht dafür geeignet, spontan angeschaltet zu werden.

Der MHZ19 kann die Daten per serieller Übertragung an den Mikrocontroller senden.

Zu guter Letzt, war es nötig, dem Sensor eine Möglichkeit zu geben, aktiv zu melden, wann zu viel CO₂ in der Innenluft vorhanden ist. Da die Sensoren von der Schulgemeinde jedoch angenommen werden müssen, war es jedoch nötig, eine Art der Rückmeldung zu finden, die subtil genug ist, dass sie nicht nervt und die Schulgemeinde nicht dazu brachte, die Sensoren abzuschalten. Dabei sollte sie trotzdem klar genug sein, dass sie die Schulgemeinde dazu motivierte, zu lüften. Damit schied die akustische Rückmeldung aus der engeren Auswahl aus, da sie bei den befragten Lehrerinnen und Lehrern nicht auf Begeisterung stieß. Es blieb die visuelle Rückmeldung. Damit man diese möglichst flexibel programmieren kann, entschied ich mich dazu, RGB-LEDs zu verwenden. Um aussagekräftige Animationen zu gestalten, habe ich dann LED-Streifen verwendet. Diese geben die Möglichkeit, LEDs mit geringem Löt Aufwand, einzeln anzusteuern. Die WS2812B Streifen eigneten sich dazu sehr gut, da man sie auch mit einer Haushaltsschere verkürzen kann und damit genau an die eigenen Ansprüche anpassen kann. Ein LED-Streifen reicht hierbei für genau 5 Sensoren.

Da einige Komponenten auf 5 Volt arbeiten und andere auf 3,3 Volt, habe ich unter anderem auch noch einen TTL-Wandler verbaut. Dieser kann vier Ein- und Ausgänge ansteuern. Vorteilhaft ist, dass dieses Modul sehr günstig ist. Die 5 Volt Referenzspannung erhält das Modul vom Mikrocontroller selbst, der die Eingangsspannung vom USB-Anschluss unregelmäßig weitergibt. Von dort muss dann diese Spannung an den CO₂-Sensor, den Referenzeingang des TTL-Wandlers und an den VIN-Pin des LED-Streifens gegeben werden.

Als Gehäuse dient eine Verteilerdose aus dem Baumarkt, die mit einem Cutter aufgeschnitten werden kann, um Löcher für Kabel und Luftzufuhr zu schaffen.

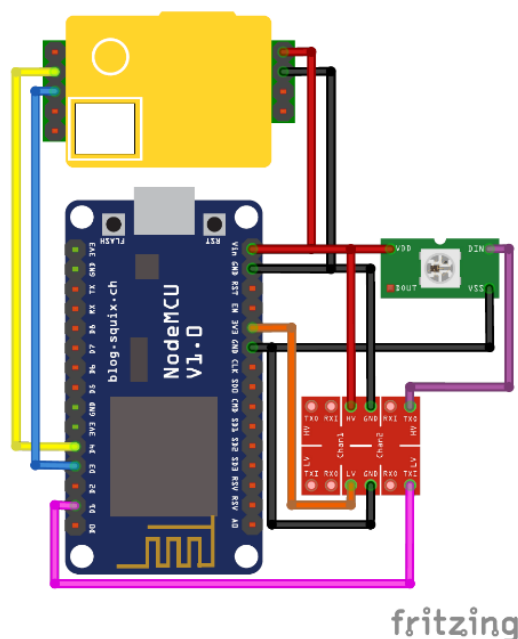
Als Stromversorgung lässt sich jedes USB-Netzteil verwenden. In unserer Schule hatte es sich jedoch als besonders praktisch erwiesen, dass Smartboards mit einem freien USB-Port vorhanden waren. Diese schalten sich mit dem Smartboard ein und auch wieder aus. Damit ist gewährleistet, dass während es Betriebs immer eine Person anwesend ist. Ein normales USB zu micro-USB Kabel kann hier verwendet werden.

Um eine Übersicht zu gewinnen, habe ich einen Schaltplan mit der freien Software „fritzing“⁵ erstellt. Diesen habe ich möglichst bildlich gehalten, um ihn später im Schulprojekt mit den Schülerinnen und Schülern verwenden zu können.

Der Aufbau ist hier in einer Schritt-für-Schritt Anleitung kurz dargestellt und ist sonst dem Schaltplan und den eingefügten Bildern zu entnehmen:

1. Anlöten der Pins an alle Platinen
2. Verbinden der Pins nach dem Schaltplan
3. Schneiden eines passenden Lochs für den CO₂-Sensor in die Verteilerdose
4. Einkleben der Komponenten und Anbringen des selbstklebenden LED-Streifens auf der Außenkante
5. Aufspielen der Software mit vorher einprogrammierten WLAN-Anmeldedaten

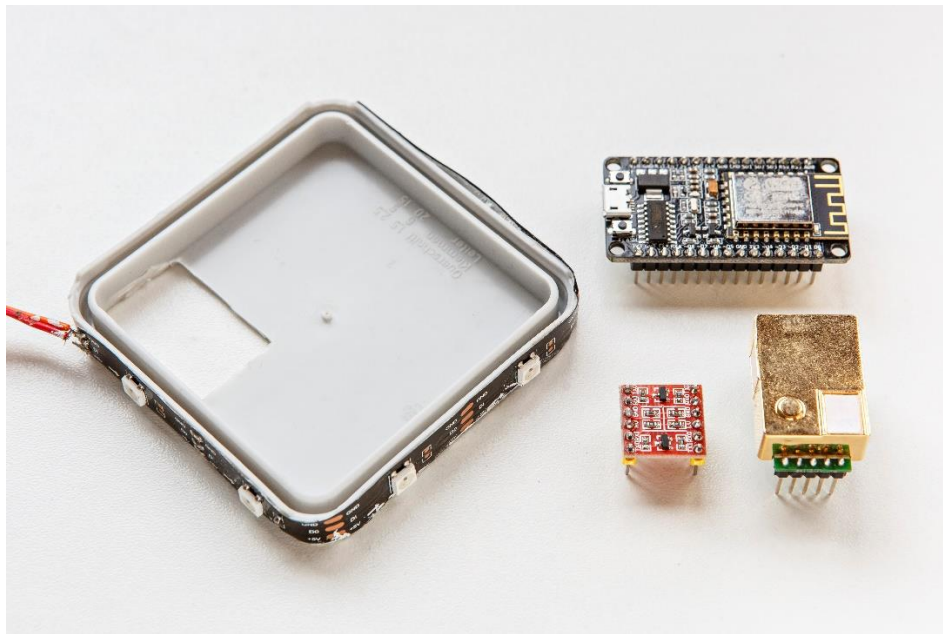
Schaltplan:



Die LED-Leiste, der Sensor und der TTL-Wandler werden mit 5 Volt aus dem VIN-Pin des ESP8266 verbunden (Rot). Das gleiche gilt für GND (Schwarz). Orange stellt hier die 3,3 Volt Referenzspannung für den TTL-Wandler (Rote Platine) dar. Die rosa markierte Leitung wird im TTL-Wandler auf die benötigten 5 Volt geregelt und verlässt diesen als violettes Kabel. Die LED-Leiste funktioniert nur mit 5 Volt Signal und Versorgungsspannung. Das gelbe Kabel stellt die RX Datenleitung dar, wobei das blaue Kabel die TX Datenleitung darstellt.

⁵ Quellenverzeichnis: 6 c). Eintrag

Einzelteile:



Hierbei sind die Bauteile: Der ESP8266 (oben rechts), der MHZ19 (unten rechts), der TTL-Wandler (unten mittig), die LED-Leiste (links)

4.2. Software

Um den Microcontroller zu programmieren, braucht man die Arduino-Umgebung (IDE)⁶. Diese lässt sich auf jedem PC mit einem der drei gängigen Betriebssysteme installieren und nutzen.

Die Anforderungen, die die Software erfüllen musste, habe ich in einer Liste zusammengetragen:

- Einfachheit und Anpassbarkeit
- Daten an einen Server zu senden, der diese speichern und wenn möglich auswerten kann
- Rückmeldung der von schlechter Luft in einfacher Form an die Schulgemeinde
- Die Möglichkeit zu bieten, die Software per WiFi zu updaten
- Sich mit einem WLAN zu verbinden

Nach kurzer Recherche fand ich den IoT-Dienst „Thingspeak“^{7 8}, bei dem man in einer kostenlosen, abgespeckten Version die Möglichkeit hat, alle 15 Sekunden Daten an einen sogenannten Channel zu schicken. In diesem Channel werden die Daten dann gespeichert und können einfach als .csv Datei exportiert werden. Außerdem gibt es auch eine rudimentäre Visualisierung der Daten. Die Daten werden zeitlich zugeordnet abgespeichert. Des Weiteren besteht auch die Möglichkeit, die Daten per MATLAB zu verarbeiten. Diese Channel sind über eine Website öffentlich abrufbar und lassen es zu, die Daten auch von Schülerinnen und Schülern einsehen zu lassen. Es ist bei der kostenlosen Version maximal möglich, vier Sensoren anzuschließen, weswegen man schnell an Grenzen gelangt. Es ist jedoch auch möglich, die Thingspeak Software auf einem eigenen Server bereitzustellen und damit alle Regulationen zu umgehen.

⁶ Quellenverzeichnis: 6 a). Eintrag

⁷ Quellenverzeichnis: 6 b). Eintrag

⁸ Quellenverzeichnis: 8. Eintrag

Für diesen Dienst existiert auch eine Bibliothek⁹, mit deren Hilfe es dem Endanwender einfach gemacht wurde, die Daten zu verschicken. Die zugehörigen Befehle ließen sich aus der Dokumentation herleiten.

Auch für den CO₂-Sensor gibt es eine Bibliothek¹⁰, die aber leider unfertig war. Deshalb musste ich einige Dateinamen umbenennen. Nachdem ich den Code und die Dokumentation des CO₂-Sensors gelesen hatte, funktionierte diese jedoch ohne Probleme. Gewisse Befehle (zum Beispiel der Check, ob der Sensor schon aufgewärmt war (`mh19.isWarming()`)) funktionierten nicht, ließen sich aber umgehen. Denn: nach einigem Testen stellte sich heraus, dass der Sensor immer den Wert „410“ sendet, solange er nicht aufgewärmt ist.

Die LED-Leiste lässt sich über die Bibliothek von Adafruit¹¹ ansteuern und dient dazu, die Schulgemeinde auf einfache Art und Weise auf schlechte Luft aufmerksam zu machen. Die programmierten Animationen sollten zu jedem Zeitpunkt laufen, weswegen es nicht möglich war, nach jedem Senden der Daten, einfach 15 Sekunden zu warten. Eine Animation wird aktiviert, wenn der zugehörige Schwellenwert erreicht ist. Dabei habe ich mich entschieden, einfache, klare Animationen einzusetzen: Ein leuchtendes Rot aller LEDs als Warnfarbe soll die Schulgemeinde aufmerksam werden lassen. Ein blinkendes Rot verstärkt dies noch. Alle Animationen, die einen Status anzeigen, sind grün und zeigen damit die Bereitschaft des Systems. Wenn der CO₂-Konzentration im akzeptablen Bereich ist, zeigt der Sensor keine Farbe an, sondern blinkt alle 60 Sekunden kurz grün, um den Status anzuzeigen.

Des Weiteren nutzte ich die Bibliothek `ArduinoOTA.h`¹², die dem Anwender die Möglichkeit gibt, die Software über WiFi zu updaten. Dabei muss sich der PC im gleichen WiFi-Netzwerk befinden wie der Arduino. Diese Bibliothek war mir aus früherer Erfahrung schon bekannt. Dabei ist es auch möglich, ein Passwort zu setzen, was vor unbefugtem Zugriff schützt. Für die Bibliothek ist es nötig, Python 2.7 auf dem Rechner zu installieren, der das Programm sendet. Leider ist die Funktion, die das Update initiiert, zeitkritisch. Man muss sie periodisch aufrufen, weswegen man nie zu lang warten darf, bevor man sie erneut aufruft.

Außerdem musste noch die Wifi-Verbindung initialisiert werden, was sich als sehr schwierig herausstellte. Denn der ESP8266 lässt sich nur sehr schwer mit Enterprise-Wifis verbinden, die an Schulen häufig zu finden sind. Der Hersteller des Mikrocontrollers „Espressif“ hat hier seine API noch nicht vollständig ausgearbeitet und wird es wohl auch nicht mehr tun. Nach vielen Stunden des Lesens der offiziellen Dokumentation der API und mehreren Fehlschlägen wandte ich mich an meinen Physiklehrer/Netzwerkadministrator, der die Möglichkeit hatte, ein WPA2-PSK Netzwerk zu erstellen. Mit diesem war es möglich, den Sensor mit dem WLAN zu verbinden¹³. Es stellte sich außerdem heraus, dass der nächstbessere Controller (ESP32) einfacher mit Enterprise-Netzwerken zu verbinden ist. Dieses Problem verschlang einige Wochen an Zeit.

Meine Vorgehensweise bestand daher vor allem darin, eine Software zu gestalten, die ohne Pausen auskommt und damit in Echtzeit reagieren kann. Kein Prozess sollte sich behindern. Dies ist wichtig, da Microcontroller immer nur einen Prozess gleichzeitig abarbeiten können. Daher war die oberste Prämisse nach einer Aktion, nicht zu warten, sondern für jede Aktion zu überprüfen, ob „die richtige

⁹ Quellenverzeichnis: 5 e). Eintrag

¹⁰ Quellenverzeichnis: 5 a). Eintrag

¹¹ Quellenverzeichnis: 5 b). Eintrag

¹² Quellenverzeichnis: 5 c). Eintrag

¹³ Quellenverzeichnis: 5 d). Eintrag

Zeit“ gekommen war. Dies bedingt sich durch die Limitationen, die durch die Animationen, die OTA-Bibliothek (Over the Air) und den Thingspeak-Server gegeben waren.

Um Übersicht zu bewahren, habe ich mehrere Funktionen deklariert.

- In `check()`; wird zuerst geprüft, ob das Intervall von 16 Sekunden vergangen ist, sodass neue Daten gesendet werden können. Diese werden dann ausgelesen und an Thingspeak verschickt. Außerdem wird der Timer wieder auf 0 gesetzt, sodass weitere 16 Sekunden gewartet werden können, bis wieder Daten an Thingspeak gesendet werden können. 16 Sekunden waren nötig, da Thingspeak das erneute Senden von Daten auf minimal 15 Sekunden limitiert.
- In `Wifi()`; wird das Wifi initialisiert. Dabei hatte ich zuerst in Betracht gezogen, das Schulische Enterprise-Wifi zu nutzen, doch später WPA2-PSK (personal shared key) genutzt. Diese Funktion wird nur einmal aufgerufen und ist daher nicht zeitkritisch. Wird jedoch kein WiFi gefunden, bleibt der Controller hier stecken. Es muss noch eine Möglichkeit gefunden werden, einen Timeout zu definieren, der die Funktion des Sensors auch bei fehlendem Netzwerk gewährleistet. Trotzdem muss gewartet werden, da der Controller eine kurze Zeit braucht, um sich mit dem Netzwerk zu verbinden.
- In `loop()`; werden die Animationen ausgeführt, die alle darauf basieren, möglichst keine Zeit zu verbrauchen. Das heißt: Bei jeder Aktion wird geprüft, ob die Animation gerade verändert werden soll oder nicht. Die Animation wird mit einem `pixels.show()`; geupdatet. Vorher werden alle Animationsdaten der Bibliothek übergeben. Zu den Animationen gehören:
 - Eine blaue Status LED, die alle 60 Sekunden einmal blinkt
 - Ein rotes Leuchten aller LEDs, wenn der CO₂-Konzentration über einen bestimmten ersten Schwellenwert steigt.
 - Ein rotes Blinken, wenn der CO₂-Konzentration über einen bestimmten zweiten, höheren Schwellenwert steigt.
 - Ein grünes Blinken, wenn der Sensor angeschaltet wird. Dies geschieht, um zu signalisieren, dass der Sensor bereit ist, zu messen.

Ein grünes Leuchten war angedacht, um den Schülerinnen und Schülern mitzuteilen, dass der CO₂-Konzentration genug gesunken ist, wurde jedoch von der Lehrerschaft als zu ablenkend empfunden und daher nicht umgesetzt.

Ein `delay()`; von 10 Millisekunden wird aufgerufen, da eine höhere Aktualisierungsrate nicht gefordert ist und der Prozessor nicht unnötig laufen muss.

4.3. Nachbau der Sensoren in einem selbstgeleiteten Schulprojekt

Um nachzuweisen, dass die Sensoren einfach nachzubauen sind, habe ich ein Schulprojekt veranstaltet, in dem 5 Schülerinnen und Schüler die Sensoren nachgebaut haben. Die Elternspende unserer Schule hat die Kosten, die sich für 5 Sensoren auf rund 170 Euro beliefen, dankenswerterweise zur Verfügung gestellt. Die Bauteile habe ich bei Banggood, einem internationalen Online-Händler, bestellt. Sie sind jedoch auch mittlerweile (September 2020) in Deutschland für einen geringeren Preis zu erhalten.

Um die Schülerinnen und Schüler vorzubereiten, habe ich allen den Schaltplan zur Verfügung gestellt. Des Weiteren hatten sie die Möglichkeit, einfache Lötaufgaben zu üben, indem sie zuallererst eine Platine mit allen mit 5 Volt belegten Kabeln verbanden. Diese konnte nicht zerstört werden und hatte damit einen guten Übungswert. Außerdem habe ich noch die Pins vorher an die CO₂-Sensoren gelötet. Dies verhinderte die Zerstörung dieser. Nach drei Projekttagen mit jeweils 4 Stunden Zeit waren die Sensoren fertig und wurden sogar von ein paar besonders kreativen Schülerinnen bemalt.

Es war zu keinem Zeitpunkt die Hilfe einer Lehrkraft vonnöten. Während dieser Zeit konnte ich mich auch um die letzten Fehler der Software kümmern. Wichtig ist in Schulen, dass die Arduino-Treiber auf den Rechnern installiert sind. Ich musste mir mit einem Laptop abhelfen, da ich keinen Admin-Zugang besaß. Zu guter Letzt war es notwendig, die Software auf die Sensoren zu spielen und die MHZ19-Module zu kalibrieren. Dies funktionierte alles ohne Probleme. Das Endresultat bestand aus 5 funktionierenden CO₂-Sensoren.

4.4. Sammeln von Daten

Nachdem das Problem des Verbindens mit dem WLAN endlich gelöst war, konnte ich vier Sensoren innerhalb der Schule installieren, die alle an ein Smartboard angeschlossen waren. Um der Schulgemeinde erklären zu können, was es mit diesen Sensoren auf sich hat, schrieb ich auch noch einen kurzen, erklärenden Brief, den ich auf jedes Lehrerpult legte, in dessen Raum sich ein CO₂-Sensor befand. Dieser befindet sich im Anhang auf GitHub. Die Sensoren ließen sich recht einfach installieren, indem ich sie in den USB-Port des Smartboards steckte (von hier aus sind die Sensoren



Sensor auf dem Beamer des Smartboards

gut sichtbar):

Meine Raumauswahl begründet sich so:

- Zwei Physik-Räume, die von allen Klassenstufen benutzt werden und von naturwissenschaftlich geprägten Lehrerinnen und Lehrern betreut werden. (43 & 46)
- Ein Raum, der vor allem von Schülerinnen und Schülern der älteren Klassenstufen benutzt wird und daher diese Altersgruppe repräsentiert. Er wird fächerübergreifend genutzt. (72)
- Ein Raum, der vor allem von Schülerinnen und Schülern der

jüngeren Jahrgänge genutzt wird und daher diese Altersgruppe repräsentiert. Auch dieser Raum wird fächerübergreifend genutzt. (34)

Mein Datensammeln teile ich in zwei Teile: Menschliches Feedback und tatsächliche Rohdaten, die von den Sensoren zurückgegeben wurden.

1. menschlich:

Da sich nach einigen Tagen herausstellte, dass in den Räumen ohne Physiklehrer das Smartboard weniger genutzt wurde als in denen mit solchem, bat ich meine betreuende Lehrerin, doch bitte eine Nachricht an die Lehrerschaft an das Whiteboard zu schreiben, die von ihr unterzeichnet war.

2. Daten aus den CO₂-Sensoren über Thingspeak:

Zu Anfang des Messzyklus, der rund zwei Wochen dauerte und vor den Weihnachtsferien begann (05.12.2019), wurden alle Sensoren auf einen Grenzwert für ein leuchtendes Rot ab 1300 ppm eingestellt. Das Blinken startete ab 2300ppm. Alle 16 Sekunden wurde ein Datenpaket mit der aktuellen CO₂-Konzentration vom Sensor an Thingspeak gesandt. Dabei wurde jedem Sensor ein Kanal zugeordnet. Insgesamt gab es vier Kanäle. Am 13.12.2019 kam ich jedoch auf die Idee, in einem der stärker frequentierten Räume die Rückmeldung vollständig auszuschalten, um zu vergleichen, wie es sich auswirkt, wenn der CO₂-Sensor nur misst und ob die Rückmeldung überhaupt

effektiv ist. Also schaltete ich in einem der Physikräume (Raum 46) die blinkende Rückmeldung vollständig aus und ließ den Sensor weiter Daten sammeln. Da hier jedoch die Testpersonen zuerst mitbekommen hatten, dass sie lüften sollten, sobald der Sensor blinkt, erwies sich diese Reihenfolge als nicht repräsentativ. Die Personen hätten glauben können, dass der Sensor defekt sei oder die zum Lüften nötige CO₂-Konzentration noch nicht erreicht sei. Zur Korrektur dieses Fehlers schaltete ich die Rückmeldung des Sensors am 09.02.2020 wieder ein und sammelte die danach entstandenen Daten, um sie mit denen aus der Zeit ohne Rückmeldung zu vergleichen. Die ursprüngliche Idee, Räume mit und ohne Rückmeldung zu vergleichen, verwarf ich, da bei verschiedenen Räumen zu viele unbekannte Variablen vorherrschen, sodass ich diese nicht miteinander vergleichen konnte.

Um die Daten später auswerten zu können, schrieb ich auch noch ein Python-Skript, das dazu diente, aus der .csv Datei alle Werte in einem bestimmten Uhrzeitfenster in eine Datei zu schreiben. In den Daten sind damit alle ersten, zweiten usw. Stunden aus dem gesamten Messzeitraum nach Tagen getrennt.

4.5. Materialliste

Hier noch einmal eine vollständige Liste mit Materialien, die für den Bau, Betrieb und das Auslesen der CO₂-Sensoren notwendig war:

Software:

- Arduino IDE
- Thinkspeak Account

Werkzeuge für den Bau:

- Lötkolben/Zinn und sonstige Lötausrüstung
- Cutter
- Computer mit USB-Anschluss und Netzwerkverbindung

Materialien für den Nachbau:

- MHZ19 Modul 0-5000ppm (20€)
- ESP8266 Microcontroller (3-6€)
- Female connector cable (0,5€)
- 3,3V-5V TTL Wandler (0,5€)
- Möglichst langes USB-Kabel (2€)
- Verteilerdose aus dem Baumarkt (1€)
- WS2812B LED-Streifen (1 Meter reicht für genau 5 Sensoren aus) 9€/5 -> ungefähr 2€ pro Sensor

Die Gesamtkosten belaufen sich theoretisch auf: 29 Euro

5. Ergebnisse

Ich behandle hier zum einen die Rohdatensammlung und die Erfahrungswerte, die ich beim Betrieb der Sensoren gewonnen habe. Zum anderen werden auch die Eindrücke, die ich beim Nachbau der Sensoren in dem Schulprojekt erfahren habe, beschrieben.

5.1 Gesammelte Daten zur Validierung der Nützlichkeit der Sensoren

Hier wollen wir feststellen, ob sich zweifelsfrei nachprüfen lässt, ob das Anbringen der Sensoren einen Einfluss auf die Luftqualität hat. Dies sollte ursprünglich durch einen Vergleich zwischen Räumen mit CO₂-Sensor und Räumen ohne diese geschehen, wurde aber verworfen, da hier zu viele unbekannte Variablen auftreten konnten: Die Klassenraumgröße könnte theoretisch schon einen Einfluss auf die absolute CO₂-Konzentration haben und ließe damit keinen Vergleich zwischen Räumen zu. Ein wissenschaftliches Experiment sollte immer nur eine Variable verändern.

Auch die zweite Methode, die Messdaten der Sensoren von vor und nach der Abschaltung der Rückmeldung zu vergleichen, verfehlte die Fragestellung, ob die Installation eines Sensors und die Rückmeldung durch Blinken eine Verbesserung der Luftqualität hervorrief. Im Nachgang zum ersten Abgabetermin habe ich am 09.02.2020 die Rückmeldung der Sensoren jedoch wieder angeschaltet und kann später die richtige Reihenfolge miteinander vergleichen.

Es wurde nun, wie im vorigen Abschnitt beschrieben, die LED-Leiste des in Raum 46 befindlichen CO₂-Sensors am 13.12.19 vollständig abgeschaltet, sodass nicht zu erkennen war, ob die CO₂-Konzentration zu hoch war.

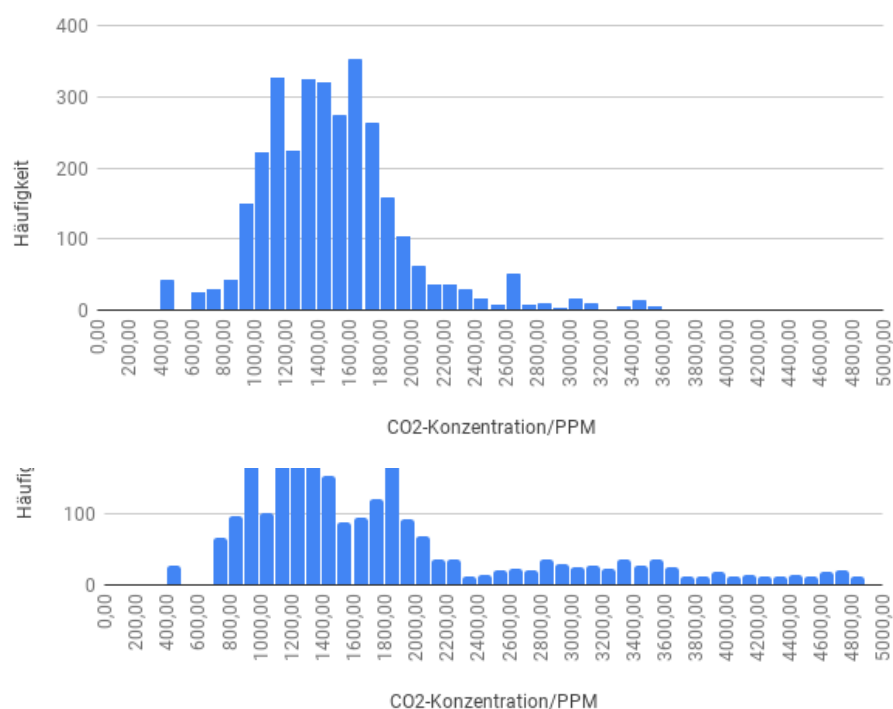
Im Messzeitraum vom 05.12.19 bis zum 20.12.19 sah die Verteilung der Daten dann so aus:

- Raum 34 hatte rund 2900 Messwerte gesammelt
- Raum 72 hatte rund 3250 Messwerte gesammelt
- Raum 43 hatte rund 5400 Messwerte gesammelt
- Raum 46 hatte rund 5800 Messwerte gesammelt

Vergleich der CO₂-Konzentration vor dem Abschalten des Blinklichts und nach dem Abschalten desselben:

Raum 46 05.12.-13.12.2019

Messwerte: 3193



Hier zeigt sich, wie sich die Verteilung verändert. In Raum 46 gibt es nach der Abschaltung der Rückmeldung des Sensors wesentlich mehr Messwerte im Bereich knapp unter 5000 PPM. Die Verteilung ist grundsätzlich flacher.

Wie in Punkt 4.4 beschrieben, stellte diese Vergleichsmethode nicht die korrekte Versuchsreihenfolge dar.

Daher vergleichen wir nun die Werte des Sensors von vor und nach dem Wiederanschalten der Rückmeldung des Sensors am 09.02.2020. Insgesamt hatte der Sensor in Raum 46 bis zum 24.02.2020 rund 20.000 Messergebnisse gesammelt. Es ist nun angemessen, jeweils den Mittelwert aus dem Zeitraum mit und dem Zeitraum ohne Rückmeldung zu vergleichen.

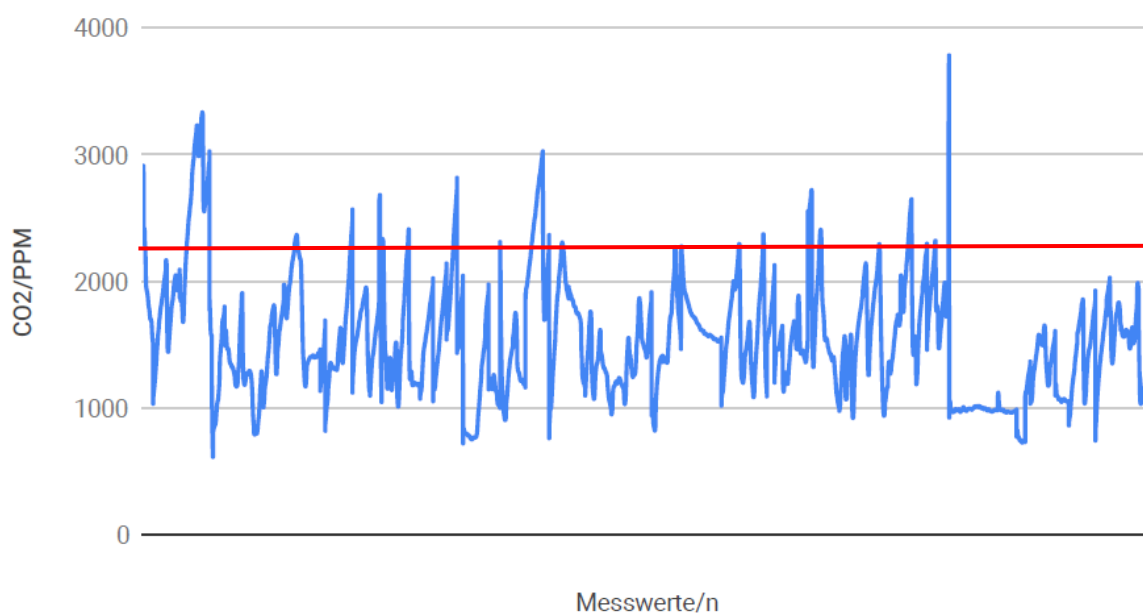
Die Messdaten sind, da der Sensor an das Smartboard gekoppelt ist, immer aus Unterrichtszeiten, da die Smartboards und damit die Stromversorgung der Sensoren nur während des Unterrichts angeschaltet sind.

Für Raum 46 ergeben sich also im Zeitraum ohne Rückmeldung (13.12.19-09.02.20) rund 1738 PPM an durchschnittlicher CO₂-Konzentration (rund 12000 Messwerte). Nachdem die Rückmeldung wieder eingeschaltet wurde ergibt sich aus dem Gesamtdurchschnitt aus dem Zeitraum vom 09.02.20 bis zum 24.02.20 (rund 5000 Messwerte) eine durchschnittliche CO₂-Konzentration von 1382 PPM. Durch die Rechnung: $((1738\text{PPM}-1382\text{PPM}) / 1738\text{PPM}) * 100 = 20,5$ ergibt sich, dass die CO₂-Konzentration in der Messperiode vom 09.02.20 bis zum 24.02.20 um 20,5 Prozent geringer ausfällt als in der vorherigen vom 13.12.19 bis zum 09.02.20. Der Absolute Unterschied beträgt -356PPM.

Überprüfen charakteristischer Muster:

Auch möglich ist, zu überprüfen, ob charakteristische Muster vorliegen. Dies habe ich bei Raum 43 getan, wo keine Änderung von Variablen wie der Rückmeldung vorgenommen wurde. Der Messzeitraum dauerte hier vom 05.12.19 bis zum 14.02.20 und beinhaltet rund 14.000 Messwerte.

Raum 43 gesamte Messdauer



Hier lässt sich beobachten, dass im Bereich von ungefähr 2300 PPM viele Einbrüche der CO₂-Konzentration vorliegen. Dies wurde durch eine Linie verdeutlicht, die den Grenzwert von 2300 PPM anzeigt.

Beobachtungen der Messaktivitäten:

Außerdem können wir dem Raumplan entnehmen, wie viele Stunden in einem Raum pro Woche gehalten werden:

- Raum 34: 41 Wochenstunden
- Raum 43: 26 Wochenstunden
- Raum 46: 26 Wochenstunden
- Raum 72: 31 Wochenstunden

Verknüpft mit den Sensordaten vom 05.12.19 bis zum 20.12.19 ergibt das für jeden Raum eine Menge von totalen Messwerten pro Wochenstunde von: (BSP: 5800 Messwerte/26 Wochenstunden = 223 für Raum 46)

- Raum 34: 70
- Raum 43: 207
- Raum 46: 223
- Raum 72: 104

Hierbei sind also die Sensoren/Smartboards in Physikräumen deutlich häufiger eingeschaltet. Mehr Messwerte wurden pro Wochenstunde erzeugt.

5.1.1 Rückmeldungen der Endnutzer

Während der Laufzeit der Sensoren wurde ich von zahlreichen Personen angesprochen, - Lehrerinnen und Lehrern und Schülerinnen und Schülern – die das Projekt befürworteten und ihrer Meinung nach mehr lüfteten. Trotzdem wurde mir in einem Fall auch die Rückmeldung gegeben, der Sensor funktioniere ja gar nicht. Man habe gewartet, bis er rot blinke, doch dies sei nie passiert. Dabei stellte sich heraus, dass das Smartboard nie angeschaltet gewesen war. (Raum 34) Dies führte dazu, dass sogar weniger gelüftet wurde. Grundsätzlich wurde auch mehr über die Thematik des Gases CO₂ gesprochen.

5.2. Erfahrungswerte vom Nachbau der Sensoren im Schulprojekt:

Während eines selbstgeleiteten Schulprojektes habe ich die beschriebenen Sensoren mit Schülerinnen und Schülern nachgebaut, wobei ich darauf geachtet habe, wie gut sie damit zurechtkommen. Die dabei gesammelten Erfahrungswerte möchte ich hier noch einmal darlegen.

Wir waren eine Gruppe aus 5 Teilnehmern, von denen drei schonmal gelötet hatten und zwei nicht. Dabei waren aber auch die Lötkenntnisse des einen eher rudimentär.

Trotzdem gelang es allen mit Löterfahrung die Sensoren nachzubauen. Die Teilnehmerinnen trauten sich jedoch nicht, auch Sensoren zu bauen, sondern beschäftigten sich damit, diese zu bemalen. Sie zum Lötten zu überzeugen, gelang trotz mehrfacher Ermutigung nicht. Die etwas erfahreneren Teilnehmer waren nach kurzer Einweisung jedoch in der Lage, die Bauteile zu verlöten und verstanden auch den Code, den ich ihnen erklärte. Schlussendlich wurden alle CO₂-Sensoren gebaut und konnten später erfolgreich zur Datenerfassung eingesetzt werden.

Der durch die Elternspende vorgegebene Preisrahmen wurde weit unterschritten. Die Gesamtkosten beliefen sich auf rund 170 Euro. Ein Sensor kostete hier rund 34 Euro. Fertiglösungen beginnen ab rund 150 Euro und können bis zu 300 Euro kosten^{14 15}.

5.3 Erfahrungswerte bei der Entwicklung der Hardware und Software

Zum einen stellte sich heraus, dass es aufgrund der schlecht entwickelten API, sehr schwer ist, einen ESP8266 mit vorhandenen Mitteln an ein WPA2-Enterprise WiFi anzumelden.

Zum anderen wurde klar, dass die Thingspeak Umgebung Limitationen aufweist, die recht drastisch sind: Es können nur maximal vier Sensoren betrieben werden, ohne dass die Softwarelimitationen überschritten werden. Denn: Jeder Sensor benötigt einen sogenannten Channel, in dem die Daten gesammelt werden. Dies betrifft zumindest die kostenfreie Version.

6. Ergebnisdiskussion

6.1. Erfahrungen aus dem Nachbau der Sensoren

Die Schülerinnen und Schüler schafften es insgesamt sehr gut, die Sensoren nachzubauen. Insofern ist die Frage, ob man einfach nachzubauende Sensoren entwerfen kann, mit Ja zu beantworten. Jedoch war es erschreckend, festzustellen, dass die Teilnehmerinnen sich nicht trauten, zu löten. Über die Herkunft dieser Ängste möchte ich nicht spekulieren. Die Teilnahme am Projekt war außerdem freiwillig.

Auch preislich sind die Sensoren im Rahmen der Anforderungen. Die Sensoren kosten mit einem Preis von 34 Euro die Hälfte dessen, was Fertiglösungen kosten. Damit sind sie für Schulen tatsächlich geeignet.

Die Anpassbarkeit hat sich als sehr gut ergeben. Während des Sammelns der Daten konnte ich mehrfach Parameter per WLAN updaten. Diese Methode war zufriedenstellend zuverlässig. Insofern habe ich damit meine anfangs gestellten Anforderungen meiner Ansicht nach vollständig erfüllt.

Der Code ist aufgeräumt genug, um die beiden Grenzwerte für die LEDs eigenhändig anzupassen und andere Animationen einzufügen.

6.2. Mess- und Erfahrungswerte

Die Rückmeldung der Lehrerinnen und Lehrer, die das Prinzip nicht verstanden hatten, führt zu dem Schluss, dass mein Text zur Beschreibung der Sensoren nicht aussagekräftig genug gewesen ist. Insofern ist es wichtig, genau zu kommunizieren, was ein Sensor tut. Beziehungsweise ein Gerät von seinen Nutzern verlangt. Das Auslegen eines Informationstextes¹⁶ war jedoch grundsätzlich hilfreich.

Grundsätzlich machen die Sensoren die Schülerinnen und Schüler jedoch aufmerksamer auf den Einfluss von CO₂ auf den Menschen. Es wurde zu einem Gesprächsthema in der Schule.

Des Weiteren lässt sich aufgrund der im Nachgang durchgeführten Untersuchungen jetzt deutlich besser sagen, dass der Einbau der Sensoren einen Nutzen hat. Nachdem die durchschnittliche CO₂-Konzentration nach dem Einschalten der Rückmeldung um rund 20,5 Prozent gesunken war, deutet dies auf eine Korrelation zwischen der Rückmeldung der Sensoren durch Blinken und der Lüftmoral der Menschen hin. Ein weiteres Indiz hierfür ist die Betrachtung des Raumes 43. Der Absolute

¹⁴ Quellenverzeichnis: 7 a). Eintrag

¹⁵ Quellenverzeichnis: 7 b). Eintrag

¹⁶ Quellenverzeichnis: 1. Eintrag

Rückgang um 356 PPM lässt darauf schließen, dass auch die Konzentrationsfähigkeit der Menschen in den Räumen signifikant angestiegen sein muss. Verbunden mit der Harvard-Studie, die für jede 400 PPM einen Rückgang der Konzentrationsfähigkeit um 21% angibt, kann man sagen, dass auch hier die Konzentrationsfähigkeit stark gestiegen sein muss.

Die charakteristischen Muster lassen darauf schließen, dass das Öffnen ein Lüftvorgang bei 2300 PPM, dem Wert, ab dem der Sensor beginnt zu blinken, beginnt. Dies scheint darauf hinzuweisen, dass die Rückmeldung der Sensoren effektiv ist und die Menschen in den Klassenräumen aufgrund des Blinkens anfangen, zu lüften.

Die in der vorherigen Methode, die Messwerte von vor und nach dem Abschalten des Sensors zu vergleichen, begangenen Methodikfehler wurden hier korrigiert. Die kurze Messdauer wurde so verlängert, dass eine relevante Menge an Messdaten verglichen werden konnte.

Das Problem, dass die Sensoren mit dem ESP8266 sich nicht mit einem Enterprise-WiFi verbinden lassen, ließe sich nur mit einem ESP32 beheben, der jedoch nicht viel mehr Geld kostet.

7. Fazit/Zusammenfassung:

Aufgrund des erfolgreichen Schulprojekts würde ich das angestrebte Ziel als erreicht bezeichnen. Die Sensoren sind günstig, einfach nachzubauen und bieten einen didaktischen Vorteil gegenüber Fertigprodukten. Mein Projekt bietet hier also ein vollständiges Konzept zum Bau von Sensoren, die auch noch beliebig erweitert werden können. Ein Einbau von Luftfeuchtigkeitssensoren oder das Abfragen der Temperatur wäre ohne weiteres möglich.

Die Effektivität der Sensoren konnte in diesem Projekt anhand der erfassten Messreihe im Nachhinein nachgewiesen werden. Mein Plan, weitere Messungen durchzuführen, war erfolgreich. Um die Ergebnisse jedoch noch weiter zu untermauern, könnten noch weitere Untersuchungen durchgeführt werden, welche, genauso wie in Raum 46 geschehen, Kontrollgruppen beinhalten. Als schwierig stellt sich außerdem dar, einzelne, unterschiedliche Räume miteinander zu vergleichen, da diese oft andere Luftvolumen und Klassengrößen aufweisen. Was auch klar festzustellen war, war, dass die Fachräume der Physik im gleichen Zeitraum das Smartboard wesentlich öfter angeschaltet hatten. Verbunden mit der Raumbelegung wurde dieser Verdacht noch verstärkt.

Darauf basierend kann der in diesem Projekt verwendete Sensor verlässlich dazu verwendet werden, die Luftqualität in Klassenräumen zu verbessern.

Die Effektivität der Sensoren könnte noch weiter verbessert werden, indem man die soziale Komponente weiter einbindet und den Informationstext¹⁷ noch verbessert. Ein direkter Kontakt mit Lehrern und Schülern lässt diese aufgeschlossener werden.

Als Ausblick würde ich außerdem vorschlagen, anstatt des ESP8266 den ESP32 zu verwenden. Dieser hat die Möglichkeit, sich auch in Enterprise-Wifis anzumelden. Der Code müsste hierzu entsprechend angepasst werden, was aber keine all zu große Hürde darstellt.

Problematisch hat sich auch erwiesen, dass Thingspeak nur vier Sensoren zulässt. Ein möglicher Workaround wäre, alle 8 Felder, die pro Channel verfügbar sind, zu nutzen und die Daten von maximal 8 Sensoren in einem Stück an Thingspeak zu senden. Dabei müssten die Daten lokal gesammelt werden, was eine kleinere Hürde darstellt. Mit diesem Trick wäre es möglich, 32 Sensoren zu betreiben. Eine andere Möglichkeit wäre es auch, die Lizenz zu kaufen oder nach einer

¹⁷ Quellenverzeichnis: 1. Eintrag

Open Source Plattform zu suchen. Thingspeak lässt es außerdem zu, die Bibliothek auch mit selbstbetriebenen Servern zu nutzen.

Nach Alternativen zu Thingspeak müsste man eher Ausschau halten. Es gibt außerdem die Möglichkeit, einen kleinen lokalen Webserver zu erstellen, der die Daten sammelt, was außerdem den Vorteil der Datenhoheit mit sich bringt. Dieser hat nur den Nachteil, dass die gute Visualisierung und die API von Thingspeak nicht übernommen werden kann.

Außerdem wurde durch die Installation der Sensoren in Kombination mit Fridays for Future zumindest in meinem Unterricht mehr über den Einfluss von CO₂ auf Mensch und Umwelt gesprochen. Dies hat zur Folge, dass sich mehr Menschen über dieses klimaschädliche Gas bewusst werden. Denn es ist vielen gar nicht klar, zu welchen Mengen sich CO₂ in unserer Atemluft befindet. Man kann sich auch unter der Einheit PPM mehr vorstellen.

Zur Erweiterung des Open-Source-Gedankens, der dieses Projekt prägt, habe ich mich an den Makerspace Darmstadt e.V.¹⁸ (eine offene Werkstatt) gewandt, der mir eine Plattform zur Weitergabe von Informationen über die Sensoren an andere Schulen geben kann und möchte. Die Schüler an diesen Schulen wären dann in der Lage, die Sensoren nachzubauen. Gespräche mit Lehrern von anderen Schulen haben gezeigt, dass Interesse an solchen Lösungen besteht. Auch in Büroräumen könnten diese Sensoren für eine Steigerung der Konzentrations- und Leistungsfähigkeit der Menschen dienen und das Wohlbefinden steigern.

¹⁸ Quellenverzeichnis: 9. Eintrag

8. Quellenverzeichnis

1. all mein Code für den ESP8266, sowie der Sortierungscode für die Daten (aus Punkt 4.4) und das Dokument, das auf jedes Lehrerpult gelegt wurde:
<https://github.com/Jasper-Sic/CO2-Sensor> (Abrufdatum: 14.01.20)
2. Studie zur Auswirkung von CO₂ auf die kognitive Leistungsfähigkeit:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4892924/pdf/ehp.1510037.pdf>
(Abrufdatum: 14.01.20)
3. Veröffentlichung des Umweltbundesamtes:
https://www.umweltbundesamt.de/sites/default/files/medien/pdfs/kohlendioxid_2008.pdf
(Abrufdatum: 14.01.20)
4. Video, das als Inspiration diente (Tom Scott und Kurtis Baute):
https://www.youtube.com/watch?v=1Nh_vxpycEA (Abrufdatum: 14.01.20)
5. Alle verwendeten Softwarebibliotheken:
 - a. MHZ19B: <https://github.com/crisap94/MHZ19> (Abrufdatum: 14.01.20)
 - b. Adafruit Neopixel: https://github.com/adafruit/Adafruit_NeoPixel (Abrufdatum: 14.01.20)
 - c. ArduinoOTA: https://arduino-esp8266.readthedocs.io/en/latest/ota_updates/readme.html (Abrufdatum: 14.01.20)
 - d. Arduino Wifi: Ist in Arduino enthalten. Daher: <https://www.arduino.cc/> (Abrufdatum: 14.01.20)
 - e. Thingspeak: <https://github.com/mathworks/thingspeak-arduino>
 - f. Installation des ESP8266:
http://arduino.esp8266.com/stable/package_esp8266com_index.json (Abrufdatum: 14.01.20)
6. Alle verwendeten Softwareumgebungen:
 - a. Arduino: <https://www.arduino.cc/> (Abrufdatum: 14.01.20)
 - b. Thingspeak: <https://thingspeak.com/> (Abrufdatum: 14.01.20)
 - c. Fritzling: <https://fritzing.org/home/> (Abrufdatum: 14.01.20)
7. Kommerzielle Sensoren:
 - a. https://www.conrad.de/de/p/tfa-dostmann-airco2ntrol-coach-co2-anzeige-co2-messgeraet-weiss-1999137.html?WT.mc_id=google_pla&WT.srch=1&ef_id=EAlalQobChMlt9_uXraG5wIViOd3Ch2aPgAoEAKYDiABEglblvD_BwE:G:s&gclid=EAlalQobChMlt9_uXraG5wIViOd3Ch2aPgAoEAKYDiABEglblvD_BwE&hk=SEM&s_kwcid=AL!222!3!318173911130!!!g!!
(Abrufdatum: 14.01.20)
 - b. https://www.voltus.de/?cl=details&anid=a09d2c40d8c21d7757b681cb32b26c43&gclid=EAlalQobChMIqNLhpf7p5wIVC5SyCh1YQAhnEAKYCSABEglFdvD_BwE
(Abrufdatum: 24.02.20)
8. Thingspeak Beispiel-Graph:
<https://thingspeak.com/channels/929543/charts/1?bgcolor=%23ffffff&color=%23d62020&dynamical=true&results=50&type=line>
9. Makerspace Darmstadt e.V.
<https://www.makerspace-darmstadt.de/> (Abrufdatum 24.02.20)

9. Unterstützungsleistungen

- Dr. Michael Balzhäuser: Physiklehrer und Tutor, Einrichtung eines WPA2-PSK WLANs für die Sensoren am LGG Darmstadt

- Miriam Schelbert: Lehrerin am LGG, Betreuung des Jugend forscht Projektes und organisatorische Betreuung des vorangegangenen Schulprojekts.
- Schülerinnen und Schüler des LGGs im Schulprojekt: Adrian Zimmer, Emma Braun, Mathilda Kügler, Florian Dubiel, Felix Göschel
- Dr. Nico Neumann: 1. Vorstand des Makerspace Darmstadt e.V, geplante Kooperation mit dem Makerspace zur Verbreitung der Sensoren an anderen Schulen.



2020 - Dieses Werk ist unter einer Creative Commons Lizenz vom Typ **Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 3.0 Deutschland** zugänglich. Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> oder wenden Sie sich brieflich an Creative Commons, Postfach 1866, Mountain View, California, 94042, USA.