

# Für bessere Luft im Klassenzimmer

Wir bauen uns ein CO<sub>2</sub>-Messgerät

von Mareen Przybylla und Stefan W. Huber

**Um Infektionszahlen zu verringern, wurden während der Covid-19-Pandemie in vielen Ländern Empfehlungen ausgesprochen, Innenräume regelmäßig zu lüften. Dabei wird häufig auch auf die Verwendung von CO<sub>2</sub>-Messgeräten verwiesen, um schlechte Luftqualität frühzeitig zu erkennen. In diesem Beitrag wird die Entwicklung und Erprobung einer Unterrichtseinheit für die Sekundarstufe I beschrieben, in der Schülerinnen und Schüler im Rahmen eines Physical-Computing-Projekts CO<sub>2</sub>-Messgeräte für ihr Klassenzimmer entwickeln und dabei Informatikkompetenzen aus den Bereichen Programmierung und Eingebettete Systeme erwerben.**

entwickeln zu lassen (vgl. z.B. Stahmer, 2020; Burger u. a., 2020; Umwelt-Campus Birkenfeld, 2020). Diese können kreativ gestaltet werden und beispielsweise als Ampel, mit Warnton oder über eine Text-Benachrichtigung Auskunft über die Luftqualität geben. Dabei lernen die Schülerinnen und Schüler grundlegende Aspekte des Entwurfs eingebetteter Systeme in einem fachübergreifenden Kontext kennen und beschäftigen sich mit verschiedenen Aspekten des Entwurfsprozesses sowie der Implementierung.

## Der Kontext

Gut belüftete Innenräume sind wichtig für das Wohlbefinden und die Gesundheit. Dies ist keine neue Erkenntnis – der Einfluss der Raumluft auf die Übertragung von Krankheiten rückt jedoch derzeit in Zusammenhang mit der SARS-CoV-2-Epidemie verstärkt in den Fokus. Schlechte Luftqualität, zum Beispiel nachweisbar über einen hohen Kohlendioxidgehalt, führt häufig zu Kopfschmerzen, Müdigkeit und Konzentrationschwächen und kann darüber hinaus aufgrund der in der Atemluft enthaltenen Aerosole (winzige Flüssigkeitspartikel), die wiederum Träger für infektiöse Viren sein können, zur Infektion mit dem Corona-Virus oder anderen Krankheiten führen (vgl. RKI, 2021, Abschnitt „2. Übertragungswege“). Experten des deutschen Umweltbundesamts raten daher, Innenräume regelmäßig zu lüften, insbesondere wenn sich viele Personen darin befinden. Es wird empfohlen, CO<sub>2</sub>-Sensoren zur raschen Erkennung der Lüftungsnotwendigkeit einzusetzen (vgl. UBA, 2020). Entsprechende Messgeräte sind momentan nicht immer leicht zu bekommen und außerdem recht teuer, daher entstand an verschiedenen Orten die Idee, mit einfachen Mitteln entsprechende Testgeräte von den Schülerinnen und Schülern selbst

## Didaktische Ziele

An der Pädagogischen Hochschule Schwyz wurde in Kooperation mit der Projektschule *Sek eins Höfe* im Spätsommer 2020 ein Unterrichtsprojekt entworfen, in dem solche Messgeräte für das Schulhaus entwickelt, programmiert und gestaltet werden sollten. Im Herbst konnte das Projekt dann im Unterricht mit Schülerinnen und Schülern einer siebten Klasse im Schulfach *Medien und Informatik* erprobt und aufgrund dieser Erfahrungen überarbeitet werden. Mit dieser Unterrichtseinheit sind die folgenden didaktischen Ziele verbunden:

- ▷ *Einfluss nehmen mit Informatik:*  
Durch die Entwicklung des CO<sub>2</sub>-Messgeräts wird bei den Schülerinnen und Schülern das Bewusstsein geschaffen, einen Beitrag zum Infektionsschutz leisten zu können. Damit wird das Selbstwirksamkeitsempfinden in Bezug auf ihre Informatikkompetenzen gestärkt.
- ▷ *Informatikkonzepte:*  
Es werden grundlegende Konzepte der Programmierung erlernt bzw. vertieft, wie Schleifen, Fallunterscheidungen, Vergleiche von Mess- und Schwellwerten. Bei der Verarbeitung von Sensorwerten und der Ansteuerung von Aktoren wird außerdem zwischen analogen und digitalen Daten sowie Ein- und Ausgaben und Verarbeitungselementen unterschieden.

## ▷ Fächerübergreifende Aspekte:

Für die Programmierung ihrer Messgeräte machen sich die Schülerinnen und Schüler mit den Grenzwerten guter bzw. schlechter Raumluft vertraut. An dieser Stelle können leicht Bezüge zu anderen Unterrichtsfächern hergestellt werden (vgl. Umwelt-Campus Birkenfeld, 2020): z.B. Physik (Sinkgeschwindigkeit und Reichweite von Tröpfchen), Biologie (Virusgröße, Bestandteile eines Virus, Einfluss von CO<sub>2</sub>-Gehalt der Luft auf Hirnaktivitäten) oder Chemie (Aufbau und Funktion eines CO<sub>2</sub>-Sensors). Aber auch gestalterische Fächer wie Kunst oder Technisches und Textiles Gestalten können einbezogen werden, indem mit dort erlernten Techniken beispielsweise die Gehäuse für die Geräte hergestellt werden. Wer Zugang zu 3-D-Druckern, Lasercuttern oder auch einer Holzwerkstatt hat, kann dieses Projekt ebenfalls gut nutzen.

▷ analoge und digitale Daten charakterisieren.

▷ Sensoren und Aktoren an die korrekten Ein- und Ausgänge (analog, digital) anschließen.

▷ ein Messgerät kalibrieren.

▷ einfache Programme zum Auslesen von Sensoren (z.B. Helligkeit, Temperatur, Gaskonzentration) und Ansteuern von Aktoren (z.B. LED, Display) entwickeln, die z.B. Schleifen, Fallunterscheidungen und Vergleiche von Mess- und Schwellwerten beinhalten.

Erweiterungen und Vertiefungen inhaltlicher Art sind in vielen Dimensionen denkbar. In der hier vorgestellten Unterrichtsreihe werden viele Aspekte des Entwurfs eingebetteter Systeme implizit behandelt, die aber auch explizit thematisiert werden könnten. Dies bietet sich vor allem an, wenn die Schülerinnen und Schüler bereits ausreichende Vorkenntnisse in der Programmierung haben, sodass andere Inhalte in den Vordergrund rücken können.

## Vorkenntnisse und Unterrichtsinhalte

Aus informatischer Perspektive steht die Gestaltung eines einfachen eingebetteten Systems im Vordergrund. Dabei können je nach Zielgruppe unterschiedliche Aspekte betont werden. In diesem Beitrag bewegen wir uns auf grundlegendem Niveau und setzen nur minimale Vorkenntnisse voraus. Es ist hilfreich, wenn die Lernenden bereits mit blockbasierter Programmierung (z.B. SCRATCH) vertraut sind. In dieser Unterrichtsreihe werden algorithmische Grundstrukturen und Variablen nicht mehr explizit eingeführt, sondern nur bei Bedarf in Erinnerung gerufen.

Stattdessen wird der Fokus auf das Thema *Physical Computing* gelegt, das am Beispielprojekt der CO<sub>2</sub>-Messgeräte vertieft wird. Beim Physical Computing geht es darum, auf kreative Weise interaktive, physische Objekte zu gestalten und entwickeln, die über Sensoren und Aktoren mit ihrer Umwelt interagieren (vgl. Przybylla, 2018). In der Unterrichtsreihe lernen die Schülerinnen und Schüler, wie man mit Sensorik Umweltdaten erfassen und auf einem Mikrocontroller verarbeiten kann, um mithilfe von Aktoren verschiedene Ausgabemöglichkeiten zu nutzen. Dabei schlüpfen sie in die Rollen von Erfindern und Entwicklern, bearbeiten Rechercheaufgaben und führen Untersuchungen im Schulhaus durch, bevor sie schließlich ihre Geräte bauen, programmieren und gestalten.

Konkret werden die folgenden Lern- und Kompetenzziele verfolgt: Die Schülerinnen und Schüler können ...

▷ Fachbegriffe korrekt verwenden, z.B. Sensor, Aktor, analog, digital, Eingabe, Verarbeitung, Ausgabe, Kalibrierung, Messwert, Referenzwert.

▷ die Bestandteile eines digitalen Systems den Kategorien Eingabe, Verarbeitung und Ausgabe zuordnen.

▷ Beispiele für Sensoren und Aktoren nennen.

## Architektur eingebetteter Systeme

Typischerweise enthalten eingebettete Systeme einen Prozessor (z.B. Mikroprozessor oder Mikrocontroller mit zusätzlicher Peripherie) und je nach konkreter Anwendung optional zusätzliche Prozessoren(en) oder anwendungsspezifische integrierte Schaltungen (ASIC) zur Erhöhung der Rechenleistung. Im Physical Computing werden oft Prototyping-Boards mit Mikrocontrollern verwendet, die zusätzliche Peripheriegeräte enthalten, um die Kommunikation mit dem Controller und dessen Programmierung zu erleichtern und einen einfachen Zugang zu den Ein-/Ausgabepins zu ermöglichen. Zusätzlich zu den Benutzungsschnittstellen gibt es Schnittstellen zu Sensoren, die Veränderungen in der Umgebung erfassen, z.B. Geräuschpegel, Bewegung oder Temperatur und Aktoren wie Leuchten, Servomotoren oder Lautsprecher, die zur ständigen Interaktion mit der Umgebung des Systems genutzt werden (vgl. Bringmann u.a.,<sup>3</sup>2018, S.4 ff.).

Im hier vorgestellten Projekt wird als „Kernstück“ ein Prototyping-Board (Arduino Uno) mit einem Luftqualitätssensor MQ135 zur Erfassung der Gaskonzentration in der Luft und verschiedenen Aktoren (z.B. RGB-LED, LCD, Summer) zur Ausgabe der Messwerte und entsprechender Warnungen verwendet.

## Datenerfassung

Bezüglich der Datenerfassung unterscheidet man im Allgemeinen zwischen zeitkontinuierlichen Systemen, die Signale kontinuierlich verarbeiten, und diskreten Systemen, die ereignis- oder zeitgesteuert sind und diskrete Signale verarbeiten, die möglicherweise durch Abtastung aus zeitkontinuierlichen Signalen gewonnen wurden (vgl. Lee/Seshia,<sup>2</sup>2017, S.44f.).

Im hier vorgestellten Projekt werden die Sensordaten zeitgesteuert erfasst, dabei bestimmen die Schülerinnen und Schüler, wie zeitnah zur Überschreitung der Grenzwerte gehandelt werden muss (siehe auch nachfolgendes Thema „Echtzeitanforderungen“). Es wer-

**Bild 1:**  
**Prototyping-Plattform**  
**Arduino Uno mit**  
**Seeed Studio Grove.**



den diskrete Werte erfasst, da der Sensor zwar über chemische Reaktionen seinen Widerstand ändert und somit ein analoges Signal liefern könnte, aber durch die Analog-Digital-Wandlung nur eine diskrete Menge an Werten erfassbar ist (Auflösung 10 Bit, also 1024 Werte).

## Echtzeitanforderungen

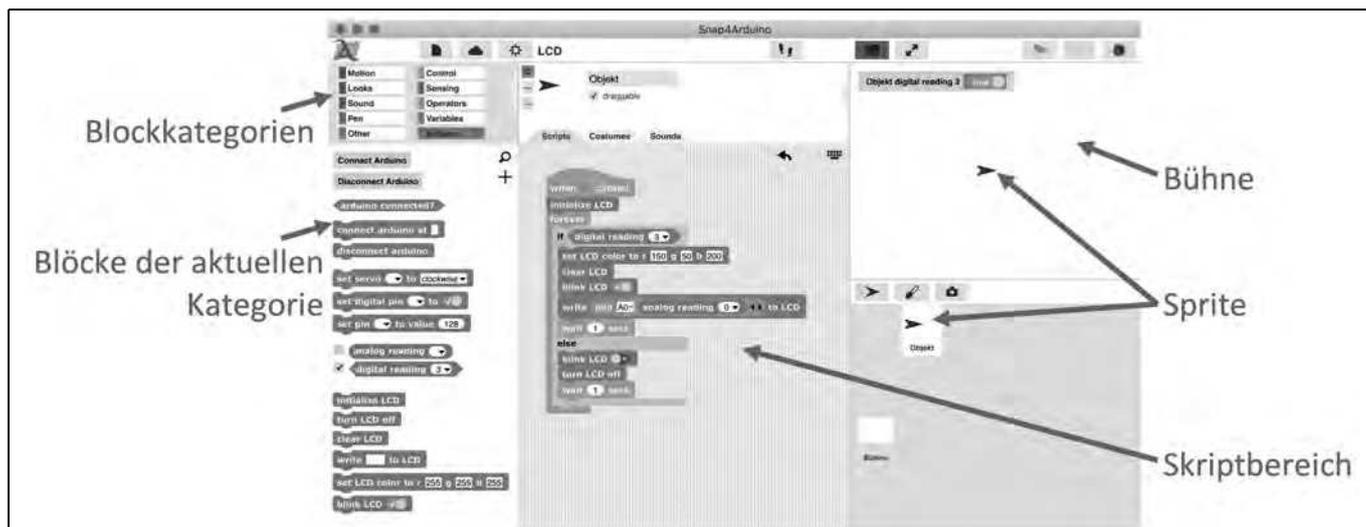
Abhängig von ihrem jeweiligen Zweck müssen eingebettete Systeme strenge Anforderungen erfüllen, z.B. Echtzeitanforderungen, um korrekte Ergebnisse in einer vorgegebenen (meist sehr kurzen) Zeit zu liefern, die mit Sicherheit vorhersagbar ist (vgl. Vahid/Givargis, 2002, S.69ff.). Dies ist insbesondere in sicherheitskritischen Bereichen relevant, z.B. bei automatisch betriebenen Zugsystemen oder in der computergestützten Chirurgie. Bei Echtzeitsystemen wird zwischen harter und weicher Echtzeit unterschieden, d.h. ob ein Überschreiten des Zeitlimits als Systemfehler angesehen wird oder bis zu einem gewissen Grad tolerierbar ist (vgl. Lee/Seshia, 2017, S.327). Während im ersten Fall ein Fehler katastrophale Folgen haben kann (z.B. das Ende von Menschenleben), beeinträchtigt er im zweiten Fall nur die Qualität des Systems (z.B. kurze Unterbrechungen eines Videostreams). In der Literatur wird manchmal noch eine dritte Kategorie von moderaten Echtzeitanforderungen beschrieben: Wenn die Frist nicht eingehalten wird, führt dies zwar nicht zu einem unmittelbaren Schaden, aber die Ergebnisse der Berechnung sind unbrauchbar (vgl. Wörn/Brinkschulte, 2005, S.321f.). Die Laufzeit kann also kritisch werden und die Funktion eines Systems beeinflussen, nicht nur die Leistung oder die Benutzungsfreundlichkeit (vgl. Lee/Seshia, 2017, S.xii).

In typischen Physical-Computing-Projekten werden harte Echtzeitanforderungen nur selten gestellt; auch im hier vorgestellten Projekt sind die Anforderungen eher weich, da leichte Verzögerungen tolerabel sind und nicht sofort zu körperlichen Schäden der Personen im Raum führen. Anders wäre es beispielsweise bei einem Rauchmelder, da eine entsprechende Rauchkonzentration binnen weniger Atemzüge tödlich sein kann. Mit den Schülerinnen und Schülern kann diskutiert werden, wie oft Sensorwerte erfasst werden sollen und welche Auswirkungen dies beispielsweise auf die Akkulaufzeit der Geräte hat, ob sich dann die Nutzung eines Netzteils anbietet, was bei Stromausfall passiert usw. (siehe auch folgendes Thema „Herausforderungen und Entwurfsmetriken“).

## Herausforderungen und Entwurfsmetriken

Der Entwurf von eingebetteten Systemen bringt einige technische Herausforderungen mit sich, die immer wieder auftreten und aus denen sich typische Methoden und Konzepte zur Problemlösung ableiten lassen. Dazu gehören Verlässlichkeit und Verfügbarkeit der Systeme, Nebenläufigkeit realweltlicher Prozesse, die in einer sequenziellen Semantik abgebildet werden müssen oder auch die Heterogenität eingebetteter Systeme, die schwieriger zu analysieren und zu entwerfen sind als homogene Systeme, da sie dynamische, physikalische Prozesse beinhalten, die nie vollständig vorhersehbar sind (vgl. Lee/Seshia, 2017, S.15). Zusätzlich gibt es bei der Entwicklung eingebetteter Systeme sogenannte Entwurfsmetriken zu berücksichtigen, z.B. Entwicklungs- und Produktionskosten, Größe, Leistung, Energie oder Flexibilität. Diese Metriken stehen in Konkurrenz zueinander (vgl. Vahid/Givargis, 2002; Bringmann u.a., 2018). So führt eine höhere Leistung oft zu einem höheren Stromverbrauch und damit zu einer geringeren Energieeffizienz. Systeme, die sehr wenig Strom verbrauchen, können möglicherweise die Echtzeitanforderungen nicht erfüllen usw. Entwickler müssen also Argumente, z.B. für und gegen schnell berechnete, aber ungenaue und langsamer berechnete, aber genaue Ergebnisse, sorgfältig abwägen und entsprechende Entscheidungen treffen (*Trade-Offs* – vgl. Wolf, 2009, S.89).

Beim Entwurf der CO<sub>2</sub>-Messgeräte soll beispielsweise beachtet werden, dass das entworfene System in jedem beabsichtigten Fall und zu jeder Zeit das gewünschte Ergebnis liefert, also morgens zu Schulbeginn im Klassenzimmer genauso, wie nachmittags in der Turnhalle, egal ob Sommer oder Winter, hell oder dunkel usw. Parallel stattfindende Prozesse sollen nicht zu völligem Stillstand (Deadlocks) führen, und die Antwortzeit des Systems auf Nutzereingaben soll möglichst gering sein. Die Schülerinnen und Schüler können in diesem Zusammenhang Überlegungen darüber anstellen, wie viel Antwortzeitverzögerung noch akzeptabel ist, ab wann es als störend wahrgenommen wird und wie man das Problem algorithmisch lösen kann. Es ist beim Entwurf der Geräte auch nicht zu hundert Prozent klar, in welchen Umgebungsbedingungen sie letztlich eingesetzt werden. Beispielsweise könnten gewisse Parameter, wie Temperatur und Luftfeuchtigkeit, mit weiteren Sensoren erfasst und einbezogen werden, andere werden bewusst vernachlässigt (bei-



**Bild 2: Programmierumgebung Snap4Arduino.**

spielsweise anders zusammengesetzte Atemluft in einem Chemieraum). Auch bezüglich der Entwurfsmetriken können die Schülerinnen und Schüler bewusste Entscheidungen treffen: In unserem Projekt haben wir zum Beispiel aus Kostengründen einen Luftqualitätssensor genutzt, statt einen deutlich teureren CO<sub>2</sub>-Sensor, was aber die Genauigkeit der Messung negativ beeinflusst – dies wird bewusst und begründet in Kauf genommen.

## Technische Vorbereitungen und Unterrichtswerkzeuge

### Verwendete Hardware zur Sensordatenerfassung

Für die Projektdurchführung haben wir uns aus pragmatischen Gründen für die Verwendung eines Luftqualitätssensors MQ135 entschieden. Dieser misst zwar nicht ausschließlich den CO<sub>2</sub>-Gehalt der Luft, sondern gibt Auskunft über die Konzentration verschiedener Gase in der Atemluft und könnte somit zur Detektion von Gaslecks oder Rauch genutzt werden, dafür ist er aber deutlich günstiger in der Anschaffung als reine CO<sub>2</sub>-Sensoren. Geht man davon aus, dass in Klassenzimmern die Konzentration anderer Giftgase gleich gering bleibt, kann man zudem den CO<sub>2</sub>-Gehalt rechnerisch ermitteln.

CO<sub>2</sub>-Messgeräte geben üblicherweise den Anteil an CO<sub>2</sub> in der Atemluft in ppm (= parts per million) an. Der hier verwendete Sensor liefert jedoch nicht diesen Wert, sondern besitzt je nach Gasteilchendichte der Luft einen anderen Widerstand, der beispielsweise mit einem Mikrocontroller ausgelesen werden kann. Zur Ermittlung der für dieses Projekt interessanten CO<sub>2</sub>-Konzentration ist es nötig, den ausgelesenen Sensorwert des Luftqualitätssensors MQ135 umzurechnen.

Aus dem Datenblatt des Sensors lässt sich die folgende Potenzfunktion ableiten:

$$\text{Konzentration}_{\text{CO}_2} = a \times (R_S/R_0)^b$$

$a$  und  $b$  sind hierbei empirisch ermittelte Parameter, die für die Messung des CO<sub>2</sub>-Gehalts mit

$$a = 116.6020682 \text{ und}$$

$$b = 2.769034857$$

angegeben werden.  $R_S$  ist der veränderliche Sensorwiderstand, über den die Gaskonzentration in der Luft bestimmt wird.  $R_0$  ist der Referenzwiderstand, der für die Kalibrierung auf einen bestimmten Wert verwendet wird. Eine genauere Erklärung zur Ermittlung dieser Werte findet sich beispielsweise im Blog von Davide Gironi (vgl. Gironi, 2014). Experimentell konnten diese Werte als gut passend und für eine Schätzung des CO<sub>2</sub>-Gehalts als ausreichend genau bestätigt werden.

Als Prototyping-Board nutzen wir den *Arduino Uno*. Für die Ausgabeseite unseres Beispielprototyps verwenden wir eine Flüssigkristallanzeige (LCD) mit RGB-Hintergrundbeleuchtung. Dies kann (und soll) natürlich im Unterricht durch die Schülerinnen und Schüler beliebig variiert werden. Generell ist es für den Informatikunterricht aufgrund der dabei entstehenden Fehlerquellen und zusätzlichen Schwierigkeiten im Allgemeinen nicht empfehlenswert, Sensoren und Aktoren mittels Litzen und Lochrasterplatten selbst zu verdrahten. Stattdessen nutzen wir im Projekt Prototyping-Kits mit vorgefertigter Sensorik und Aktorik, bei der herkömmliche Steckverbinder zum Einsatz kommen, sodass keine über die Unterscheidung von digital oder analog angesteuerten Komponenten hinausgehenden Kenntnisse der Elektronik erforderlich sind. Um eine große Vielfalt an Sensoren und Aktoren anbieten zu können, wird im hier vorgestellten Setting die weit verbreitete und vielseitige Plattform *Seeed Studio Grove* verwendet (siehe Bild 1, vorige Seite).

### Verwendete Programmierumgebung: Snap4Arduino

Die hier beschriebene Unterrichtsreihe richtet sich an Schülerinnen und Schüler der Sekundarstufe I (ab

der siebten Klasse), und es werden nur geringfügige Programmierkenntnisse vorausgesetzt. Dementsprechend sollte die verwendete Programmierumgebung möglichst intuitiv erfassbar sein und an potenzielle Vorkenntnisse aus der Primarstufe anknüpfen. Daher fiel der Entscheid auf *Snap4Arduino*, eine blockbasierte Programmierumgebung, die sich einfach mit zusätzlichen Blockbibliotheken erweitern lässt, um die gewünschten Sensoren und Aktoren verwenden zu können (siehe Bild 2, vorige Seite).

Zur didaktischen Reduktion haben wir uns für diese Zielgruppe entschieden, die Blöcke sowohl zur Kalibrierung als auch zur Messung bereits zur Verfügung zu stellen und nicht durch die Schülerinnen und Schüler selbst implementieren zu lassen, da die Hintergründe der Berechnungen kognitiv sehr anspruchsvoll sind (siehe Bild 3).

Für höhere Altersstufen wäre es aber durchaus denkbar, diesen Teil im Unterricht zu erarbeiten. Auch hier bietet es sich wieder an, fächerverbindend zu denken.

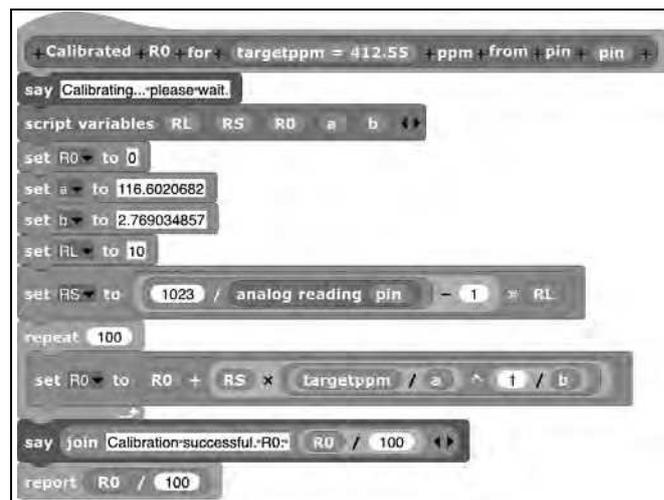
Ein Nachteil der Verwendung von *Snap4Arduino* ist, dass die Kommunikation mit dem Mikrocontroller via USB realisiert wird. Das hat zur Folge, dass das entwickelte Gerät permanent per USB mit dem Computer verbunden bleiben muss, auf dem es programmiert wird. Möchte man dies vermeiden, muss man auf eine andere Programmierumgebung ausweichen.

## Notwendige Vorbereitungen

Um die beschriebene Hardware und Programmierumgebung in Kombination nutzen zu können, sind einige Vorbereitungen nötig. Zunächst muss der Arduino mit einer angepassten Firmata-Version bespielt werden, die die Steuerung des Mikrocontrollers über USB basierend auf serieller Kommunikation übernimmt. Für Arduinos gibt es eine Standardimplementierung des Kommunikationsprotokolls Firmata. Diese mit der Arduino-IDE gelieferte Standard-Firmata genügt zwar, um die Standardbefehle in *Snap4Arduino* zu nutzen, jedoch sind für einige der verwendeten Sensoren und Aktoren Zusatzbibliotheken nötig, die in die angepasste Firmata-Version eingebunden werden und die Ansteuerung mit speziellen Blöcken seitens *Snap4Arduino* ermöglichen. Der Upload erfolgt über die Arduino-IDE. Es reicht prinzipiell, wenn die Lehrperson diese auf ihrem Rechner installiert und alle Boards einmal bespielt. Die bespielten Boards werden dann per USB mit den Schülercomputern verbunden.

Anschließend muss die Blockbibliothek mit den Erweiterungen zur Ansteuerung der verwendeten Sensoren und Aktoren in die Programmierumgebung *Snap4Arduino* importiert werden. Dies kann einfach per Drag-and-drop der XML-Datei auf den Bühnenbereich oder aber über das Menü erfolgen.

Alle benötigten Dateien sowie dazugehörige detaillierte, bebilderte Anleitungen werden am Ende des Beitrags auf der unter „Material“ angegebenen Webseite zum kostenfreien Herunterladen zur Verfügung gestellt.



**Bild 3: Aufbau (oben) und Verwendung (unten) des Kalibrierungsblocks in *Snap4Arduino*.**



## Beispielprototyp – Aufbau und Programm

Der Beispielprototyp (siehe Bild 4, nächste Seite) ist sehr minimalistisch gehalten: Wir verwenden Arduino Uno, ein Grove-Shield, ein Grove RGB-Backlight-LCD (angeschlossen an I2C) und den MQ135-Luftqualitätssensor (angeschlossen an A0). Für den Sensor muss im Vorfeld ein Grove-Anschlusskabel beschafft oder hergestellt werden.

Das Programm besteht aus zwei Teilen: der Kalibrierung und der dauerhaften Überwachung der Luftqualität. Für die Kalibrierung muss zunächst der Referenzwert in ppm angegeben werden, auf den das Gerät kalibriert werden soll. Diesen kann man entweder mit einem bereits kalibrierten Gerät bestimmen, das man direkt neben den Sensor stellt, oder aber an frischer Luft, indem man beispielsweise den auf der Webseite

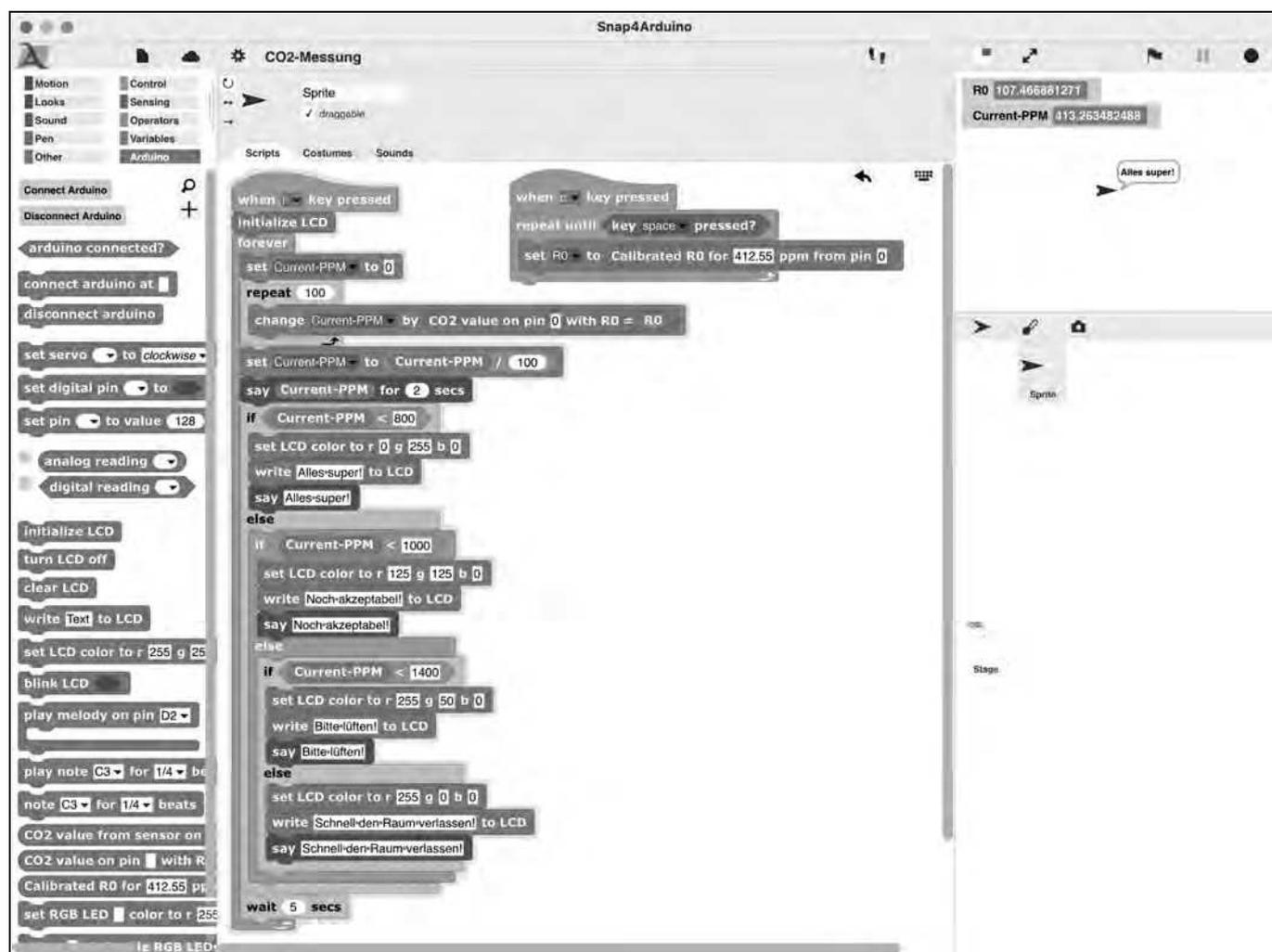
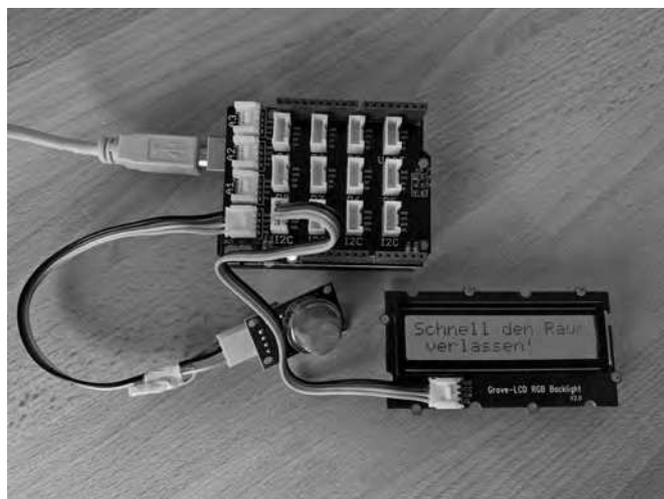
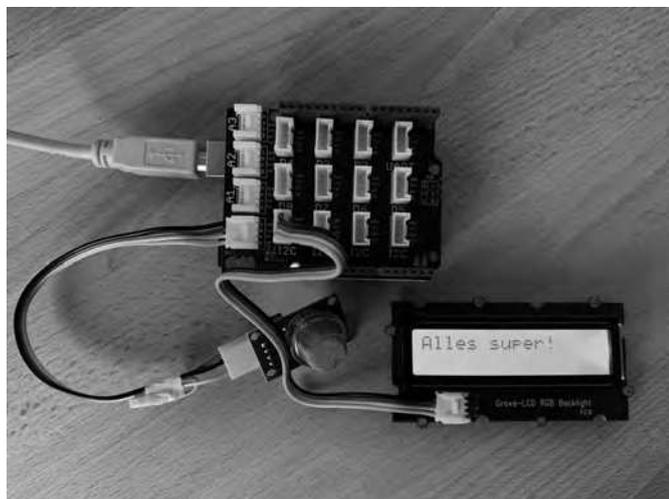
<https://www.co2.earth/>

angegebenen Wert als gültig für den eigenen Standort annimmt. Ist der Referenzwert bestimmt und eingetragen, wird die Taste „c“ (calibrate) gedrückt, anschließend sollte man die Kalibrierung laufen lassen, bis sich die Werte stabilisieren, und danach mit der Leertaste den Kalibrierwert abspeichern. Grundsätzlich ist zu beachten, dass die Sensoren sich erst einbrennen müssen und laut Herstellerangaben zunächst 24 bis 48 Stunden dauerhaft laufen sollten. Unsere Er-

fahrungen zeigen, dass die Sensoren aber auch nach ca. 10 bis 15 Minuten Laufzeit bereits stabile Werte liefern. Es schadet jedoch nicht, nach längerer Laufzeit einen erneuten Kalibrierungsdurchgang zu starten.

Das Programm zur Messung wird in unserem Beispiel dann mit der Taste „r“ (run) gestartet und läuft fortan kontinuierlich. In einer Dauerschleife wird die aktuelle CO<sub>2</sub>-Konzentration der Umgebungsluft be-

**Bild 4: Aufbau und Programm des Beispiel-Prototyps (ohne Gehäuse; links grün „Alles super!“, rechts rot „Schnell den Raum verlassen!“).**



stimmt. Um Ausreißern in der Messung entgegenzuwirken, wird dabei der Durchschnittswert aus 100 Messungen herangezogen. Dieser Wert wird auf der Bühne in der Programmierumgebung ausgegeben und anschließend ausgewertet. Je nach CO<sub>2</sub>-Konzentration erfolgt dann eine Ausgabe auf der Bühne und dem Display. Das Display zeigt zudem mit seiner Hintergrundfarbe an, wie gut die Luft zu interpretieren ist (von gut nach schlecht: grün, gelb, orange, rot). Nach fünf Sekunden Pause springt das Programm an den Beginn der Schleife zurück, und die nächste Messung wird durchgeführt.

## Aufbau der Unterrichtsreihe

Die Unterrichtsreihe gliedert sich in drei Phasen: Zunächst werden in den ersten Lektionen die allgemeinen Grundlagen im Bereich Physical Computing geschaffen. Anschließend setzen sich die Schülerinnen und Schüler vertiefend mit der Messung der CO<sub>2</sub>-Konzentration in der Luft sowie den technischen Aspekten der Entwicklung eigener CO<sub>2</sub>-Messgeräte auseinander. Dies mündet dann in Projektarbeit, in der diese Geräte durch die Lernenden erstellt, programmiert und gestaltet werden.

Im konkreten Szenario wurde die Unterrichtseinheit in neun Lektionen à 45 Minuten durchgeführt, wobei es sich jedoch empfiehlt, etwas mehr Zeit einzuplanen und – wenn möglich – in Doppellektionen zu unterrichten (oder bereits gefestigtere Grundkenntnisse der Programmierung sowie im Umgang mit dem Computer vorzusetzen).

*Die Lektionen im Überblick:*

- ▷ Lektion 1: Einführung, erster Einblick in Hardware und Software (Demo/Tutorial).
- ▷ Lektion 2 und 3: Stationenarbeit.
- ▷ Lektion 4 und 5: CO<sub>2</sub>-Messungen, Recherchearbeit, Kalibrierung von Geräten.
- ▷ Lektion 6 und 7: Projektarbeit.
- ▷ Lektion 8 und 9: Fertigstellung der Projekte (finales Kalibrieren, Testen, ggf. Korrigieren letzter Fehler), Abschlusspräsentationen.

### Erste Phase: Einführung in Physical Computing

Die Gestaltung der ersten Unterrichtsphase erfolgt mit Materialien, die für den Berliner Zusatzkurs *Digitale Welten* entwickelt wurden (siehe Abschnitte „Materialien“ und „Weiterführende Internetquellen“). Den Schülerinnen und Schülern wird zunächst kurz das Themengebiet vorgestellt. Sie erhalten anhand realweltlicher Beispiele eine kurze Einführung in eingebettete Systeme, Mikrocontrollerprogrammierung und Physical Computing, und es werden Fotos von beispielhaften Schülerprojekten zum Abstecken des Rahmens der Möglichkeiten gezeigt. Die Hardware und die Programmierumgebung werden mit einer kurzen Demo

oder in einem Tutorial vorgestellt, benötigte Programme werden installiert, konfiguriert und ausprobiert. Die Ziele dieser ersten Lektion bestehen darin, den Lernenden einen Einblick in das Themenfeld *Physical Computing* zu geben und sie die nötigen technischen Vorbereitungen für das Stationenlernen und die anschließende Projektarbeit treffen zu lassen.

In der sich anschließenden Stationenarbeit findet dann eine detaillierte Einführung in Physical Computing, die genutzten Werkzeuge und Bauteiltypen sowie die entsprechenden Programmelemente statt. Unterstützt durch eine Folienpräsentation wird der Ablauf erklärt. Die Lernenden arbeiten zu zweit an den verschiedenen Stationen, die Lehrkraft unterstützt, wenn nötig. Jede Schülerin und jeder Schüler erhält einen eigenen Laufzettel zum Abarbeiten der Stationen, auf dem die Ergebnisse der Stationenarbeit in Stichpunkten festgehalten und der eigene Lernfortschritt eingeschätzt werden. Nach jeder bearbeiteten Station berichten die Schülerinnen und Schüler der Lehrperson und führen ggf. Programmelemente vor. Wenn alles zufriedenstellend erledigt wurde, wird die Station auch von der Lehrperson als abgeschlossen gegengezeichnet. Je nach Vorkenntnissen sollten hierfür mindestens ein bis zwei Doppellektionen eingeplant werden.

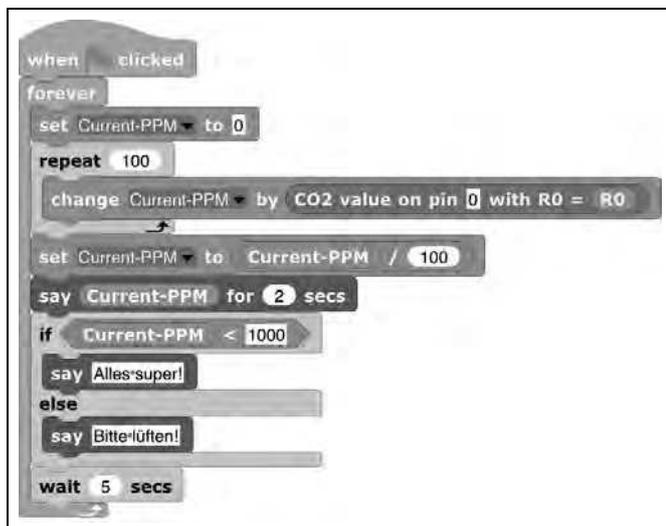
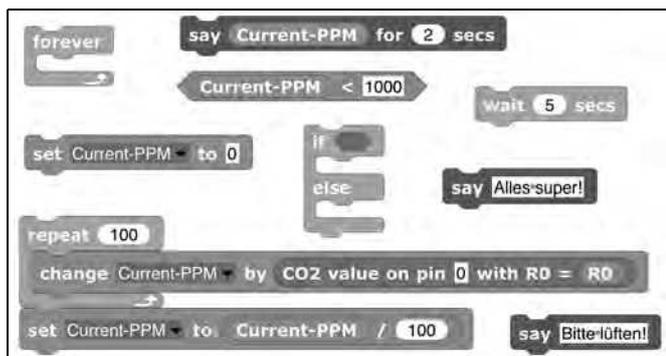
### Zweite Phase: Bestimmung der Luftqualität, technische Eigenschaften von CO<sub>2</sub>-Messgeräten

In der zweiten Phase der Unterrichtsreihe setzen sich die Lernenden vertiefend mit der Messung der CO<sub>2</sub>-Konzentration in der Luft sowie den technischen Aspekten der Entwicklung eigener CO<sub>2</sub>-Messgeräte auseinander. Hierzu recherchieren sie angeleitet durch ein Arbeitsblatt zunächst anhand von Leitfragen zu verschiedenen Aspekten der Messung von Luftqualität:

- ▷ Inwiefern kann die CO<sub>2</sub>-Konzentration der Luft etwas über die Luftqualität aussagen?
- ▷ Welche Faktoren beeinflussen die CO<sub>2</sub>-Konzentration in Innenräumen besonders stark?
- ▷ Welche Auswirkungen kann schlechte Luftqualität auf Menschen haben?
- ▷ Wofür steht die Maßeinheit „ppm“ und was bedeutet dieser Wert?
- ▷ Welche CO<sub>2</sub>-Werte (ca.) misst man üblicherweise an frischer Waldluft, in Städten, in Innenräumen?
- ▷ Was sind „gute“, „akzeptable“ und „schlechte“ Werte und wann sollte man unbedingt Maßnahmen ergreifen?

Zusätzlich zu diesen Leitfragen bietet sich an, je nach Zielgruppe bestimmte Webseiten als Anlaufstellen anzugeben oder eventuell auch Material gezielt auszuarbeiten.

Parallel zu dieser Recherchearbeit erhalten die Lernenden den Auftrag, mit professionellen CO<sub>2</sub>-Messgeräten ausgestattet die Luftqualität an verschiedenen Orten in der Schule zu überprüfen. Dabei sollen sie – je nach Anzahl verfügbarer Messgeräte in der Schule – in Gruppen den CO<sub>2</sub>-Gehalt der Luft an mindestens fünf verschiedenen Orten im Schulgebäude und auf



**Bild 5: Puzzleteile und rudimentäres Grundgerüst eines Programms für ein CO<sub>2</sub>-Messgerät.**

dem Schulhof bestimmen und dabei möglichst unterschiedliche Bedingungen erfassen. Die Messergebnisse und mögliche Besonderheiten, die die Luftqualität beeinflussen könnten (z.B. viele Personen im Raum, Fenster geöffnet usw.), werden in einer übersichtlichen Tabelle dokumentiert und im Anschluss die Ergebnisse aller Gruppen zusammengeführt und unter Einbezug der Rechercheergebnisse interpretiert.

Hieraus können dann auch Schlussfolgerungen für die Projektarbeit gezogen werden: Zunächst sollte den Lernenden bewusst werden, dass regelmäßiges Lüften sinnvoll und wichtig ist, da schlechte Luftqualität beispielsweise zu Kopfschmerzen, Müdigkeit und Konzentrationsschwächen führen kann und aufgrund der in der Atemluft enthaltenen Aerosole Infektionen mit dem Corona-Virus oder anderen Krankheiten begünstigt werden können. Aus den Messungen im Schulgebäude und der Recherche erkennen sie zudem, in welchen Bereichen es besonders wichtig ist, darauf zu achten und sich ggf. durch ein entsprechendes Gerät daran erinnern zu lassen – beispielsweise in Räumen, in denen sich viele Personen befinden, die besonders klein sind, in denen viel geschwitzt oder Deodorant benutzt wird. Über die Rechercheergebnisse und die Messungen entwickeln sie ein Gefühl dafür, welche Messwerte

repräsentativ für gute, mäßige oder schlechte Raumluft stehen. Während an frischer Waldluft etwa ein CO<sub>2</sub>-Gehalt von ca. 400 ppm zu erwarten ist, sind es in Städten im Schnitt schon etwa 500 ppm. In Innenräumen werden Werte von 700 bis 800 ppm noch als gute Luft interpretiert; ab 1000 ppm wird die Luftqualität je nach Quelle bereits als grenzwertig erachtet, manche Publikationen sehen auch erst 1400 ppm oder 2000 ppm als kritisch an. In der Diskussion können die Schülerinnen und Schüler nun festlegen, welche Grenzwerte sie in ihren Projekten verwenden werden und begründen dies anhand fundierter Argumente.

Im Anschluss daran beschäftigen sich die Lernenden dann mit der Kalibrierung der Messgeräte. Hierzu wird zunächst im Klassengespräch allgemein erarbeitet, wozu eine Kalibrierung nötig ist und wie diese funktionieren kann. Dann wird das entsprechende Programm für den verwendeten Luftqualitätssensor MQ135 gemeinsam analysiert (siehe Bild 3, Seite 94).

### Dritte Phase: Projektarbeit

Die Phase der eigentlichen Projektarbeit kann je nach Vorkenntnissen und verfügbarem Zeitrahmen unterschiedlich frei gestaltet werden und, wie oben beschrieben, auch unterschiedlich detailliert verschiedene Herausforderungen im Design eingebetteter Systeme aufgreifen. Es ist auch denkbar, das Projekt mit Elementen agiler Methoden zu unterstützen (vgl. Brichzin u.a., 2019) und somit insbesondere die Entwicklung von frühen Prototypen zu fördern, was den Ideen des Physical Computing entgegenkommt.

In der hier geschilderten Umsetzung wird insbesondere aufgrund der knapp bemessenen zeitlichen Ressourcen ein Vorgehen gewählt, indem die Schülerinnen und Schüler angeleitet durch ein hilfestellungsgebendes Video in Zweiergruppen ihre Programme für die eigenen CO<sub>2</sub>-Messgeräte erstellen. Dabei werden ihnen im Sinne des Scaffoldings die Bestandteile eines rudimentären Beispielprojekts erklärt (siehe Bild 5), d.h. der Lernprozess wird durch die Bereitstellung einer ersten Orientierungsgrundlage in Form bestimmter Hilfestellungen unterstützt, und sobald die Lernenden fähig sind, eine bestimmte Teilaufgabe eigenständig zu bearbeiten, entfernt man dieses „Gerüst“ schrittweise wieder. Die Aufgabe der Lernenden ist hier, das Beispielprojekt zu durchdringen und mit ihren eigenen Zielvorstellungen dann in ein funktionsfähiges Programm für ihr eigenes Gerät zu überführen, das neben dem Luftqualitätssensor mindestens zwei verschiedene Aktoren enthält und somit auf unterschiedliche Weise signalisiert, wie die aktuelle Umgebungsluft zu beurteilen ist und ob es bereits notwendig ist, den Raum zu belüften. Nebst der Programmierung ist auch die Aufgabe der Schülerinnen und Schüler, für ihre Geräte geeignete Gehäuse zu basteln und zu gestalten (siehe Bild 6, nächste Seite).

In der letzten Doppellektion erhalten die Schülerinnen und Schüler noch die Gelegenheit, ihren Projekten den letzten Feinschliff zu geben und die Geräte ggf. erneut zu kalibrieren. Als Abschluss der Unterrichtsreihe präsentieren sie anschließend ihre Projekte, erklären



**Bild 6: Beispielprojekte der Schülerinnen und Schüler.**

besondere Eigenschaften und reflektieren die Projektarbeitsphase.

## Lernkontrollen

Zur Überprüfung des Lernstands bietet sich an verschiedenen Stellen an, formative Lernkontrollen durchzuführen. In der konkreten Umsetzung haben wir zu Beginn der vierten Lektion mit einem Online-Test mit *Microsoft Forms* überprüft, inwiefern die Schülerinnen und Schüler in der Lage waren, die bisher kennengelernten Fachbegriffe korrekt zu verwenden, die verschiedenen Sensoren und Aktoren ihren Funktionen sowie der Eingabe und Ausgabe zuzuordnen, Unterschiede zwischen analogen und digitalen Signalen zu erfassen und entsprechende Bauteile zuzuordnen sowie einzelne Prozesse des Inputs, der Verarbeitung und des Outputs eines beispielhaften Messgeräts in die korrekte Reihenfolge zu bringen. Vor der sechsten Lektion fand eine erneute formative Lernkontrolle zur Prüfung des aktuellen Wissens- und Kompetenzstands statt, diesmal gestaltet als Quiz mit *Kahoot!* (einer kostenfreien spielebasierte Lernplattform, die von der gleichnamigen Firma betrieben wird) und insbesondere in Hinblick auf neu gelernte Konzepte und Fachbegriffe in Zusammenhang mit der Kalibrierung von Messwerten.

Zum Abschluss der Unterrichtsreihe wurden die Lernenden in einer mündlichen Prüfung zur Umsetzung ihrer Projekte befragt und sollten dabei erklären,

- ▷ welche Bauteile sie nutzten und warum.
- ▷ welche Funktion die einzelnen Befehle in ihren Programmen haben.
- ▷ warum sie bestimmte Befehle verwenden und nicht andere.
- ▷ wie sie ihre Skripte ändern könnten, ohne die Ausgabe zu verändern.
- ▷ welcher Teil des Skripts für bestimmte Funktionalitäten zuständig ist.
- ▷ welche Rolle bestimmte Variablen im Skript spielen.
- ▷ was passieren würde, wenn bestimmte Dinge geändert oder gelöscht würden.

Diese mündliche Prüfung wurde für die Bewertung der Unterrichtsreihe genutzt.

## Reflexion der Erfahrungen aus dem Unterricht

In der Umsetzung ihrer Projekte zeigten sich die Schülerinnen und Schüler kreativ: Einige nutzten beispielsweise rote und grüne LED-Lämpchen, um zu signalisieren, wann gelüftet werden soll, andere verwendeten LCDs auf denen der aktuelle Wert in ppm ausgegeben wurde, wieder andere setzten Summer ein, um auch akustisch zu warnen. Ebenfalls hat die Gestaltung der Gehäuse ihnen viel Freude bereitet, und von Joghurtbechern über Popcorn-Schachteln bis hin zu Geschenkkartons wurden viele Dinge für diesen Zweck genutzt.

Im Nachgang haben wir die Schülerinnen und Schüler in einer Online-Umfrage zu ihrer Wahrnehmung des Unterrichts sowie bezüglich ihrer Selbsteinschätzung zum Kompetenzerwerb und Wissenszuwachs befragt. Insgesamt zeigte sich ein positives Bild: Die Schülerinnen und Schüler hatten nicht nur den Eindruck, ihr konkretes Programm durchdrungen zu haben, sondern sahen sich überwiegend auch in der Lage, andere Physical-Computing-Projekte umzusetzen (genanntes Beispiel: Lichterkette aus LEDs, deren Farben regelmäßig wechseln). Sie hatten den Eindruck, viele wichtige Dinge gelernt zu haben und Programmierung und „Coden“ nun besser zu verstehen (Freitextantworten). Während die Einstiegslektionen von der überwiegenden Anzahl der Schülerinnen und Schüler noch als leicht, teilweise sogar zu leicht empfunden wurden, wurden die späteren Lektionen 4 und 5 (CO<sub>2</sub>-Messung, Recherche, Kalibrierung) bzw. 6 und 7 (Projektarbeit) als deutlich anspruchsvoller wahrgenommen. Die meisten Schülerinnen und Schüler fanden sie machbar oder schwierig, vereinzelt wurden sie auch als „zu schwierig“ empfunden. Gleichzeitig waren dies aber auch die Lektionen, die von ihnen als „am lehrreichsten“ wahrgenommen wurden. Die letzte Doppellektion zeigte dann wieder ein ausgewogenes Bild mit Ausschlägen in beide Richtungen.

Rückblickend und zusammenfassend betrachtet hat die Unterrichtsreihe gut funktioniert, allerdings mit einigen Einschränkungen:

- ▷ Wir hatten die Möglichkeit, den Unterricht in Doppellektionen abzuhalten. Dies hat uns deutlich geholfen, denn andernfalls wäre durch das Zusammensuchen der benötigten Bauteile am Stundebeginn und das Aufräumen am Ende viel Zeit verloren gegangen. Leider ist das nicht die gängige Praxis im Fach

*Medien und Informatik*, weswegen man bei einer erneuten Durchführung im normalen Unterrichtsetting mindestens 12 Lektionen einplanen sollte.

- ▷ Trotz der Doppellektionen vergingen die Unterrichtsstunden wie im Flug. Teilweise war unsere Planung zu ambitionistisch, sodass einige Dinge nicht in der Tiefe besprochen werden konnten, wie wir es gerne gehabt hätten. Auch hier gilt: Man sollte mehr Zeit einplanen oder aber die Unterrichtsschritte reduzieren.
- ▷ Einige Materialien waren für die Zielgruppe zu textlastig; es fiel auf, dass sie eher ausprobierten, als Anleitungen zu folgen, und dann Fragen auftauchten, die beim Lesen der Anleitung hätten geklärt werden können. Hier eignet sich vermutlich, knapper gehaltene Challenge Cards zu verwenden und – wo möglich – Textpassagen zu ikonisieren.
- ▷ Es passierte relativ häufig, dass die Schülerinnen und Schüler bestimmte Teile (Sensoren, Aktoren) plötzlich nicht mehr finden konnten, was den Unterrichtsablauf natürlich stört. In allen Fällen tauchten die Teile später wieder auf, meist bei denselben Personen. Hier wäre es daher sinnvoll, sich künftig eine geschickte Methode zur Arbeitsplatzorganisation zu überlegen, beispielsweise indem man auf jedem Pult eine Fächerbox bereitstellt.
- ▷ Es hat oft einiges an Zeit gekostet, den Schülerinnen und Schülern bei ganz grundlegenden Dingen zu helfen, die mit dem eigentlichen Unterricht nichts zu tun haben: Wo ist meine Datei geblieben? Wie geht Drag-and-drop? Wie kann ich das Dokument herunterladen? Neben einer besseren Organisation der Bereitstellung der Programme und Materialien (wir nutzten hierfür die Plattform *MS Teams*) wäre es vielleicht auch denkbar, ein solches Projekt erst durchzuführen, wenn die Schülerinnen und Schüler im Umgang mit ihren Computern bereits vertrauter sind und man nicht mehr so viel Unterrichtszeit für solche Dinge benötigt.

*Im Fazit:* Wir würden den Unterricht auf sehr ähnliche Weise jederzeit wieder durchführen, allerdings vermutlich erst ein Schuljahr später, da wir vermuten, dass dann viele der angesprochenen kognitiven und organisatorischen Hürden nicht mehr bestehen.

Prof. Dr. Mareen Przybylla  
Pädagogische Hochschule Schwyz  
Zaystrasse 42  
6410 Goldau  
Schweiz

E-Mail: mareen.przybylla@phsz.ch

Stefan W. Huber  
Sek eins Höfe  
Schulhaus Weid  
Weidstrasse 20  
8808 Pfäffikon  
Schweiz

E-Mail: Stefan.Huber@sekeinshoefe.ch

## Materialien

Alle verwendeten Unterrichtsmaterialien, Installationsanleitungen und benötigten Dateien sowie weiterführende Internetquellen zu den Programmierumgebungen stellen wir auf der Webseite

<https://mia.phsz.ch/Informatikdidaktik/CO2Messung>  
kostenfrei zur Verfügung.

## Literatur und Internetquellen

Brichzin, P.; Kastl, P.; Romeike, R.: *Agile Schule – Methoden für den Projektunterricht in der Informatik und darüber hinaus*. Bern: hep – der bildungsverlag, 2019.

Bringmann, O.; Lange, W.; Bogdan, M.: *Eingebettete Systeme – Entwurf, Modellierung und Synthese*. Reihe „De Gruyter Studium“. Berlin; Boston: De Gruyter Oldenbourg, 2018.

Burger, G.; Fix, R.; Gollmer, K.-U.: *Der CO2-Warner für die Schule*. In: *MAKE.*, 7. Jg. (2020) ; Heft 5, S.10–17.  
<https://www.heise.de/select/make/2020/5/2022015381334973804>

Gironi, D.: *Cheap CO2 meter using the MQ135 sensor with AVR AT-mega*. 25. Januar 2014.  
<https://t1p.de/83y1>

Lee, E. A.; Seshia, S.A.: *Introduction to Embedded Systems – A Cyber-Physical Systems Approach*. Cambridge (MA, USA): MIT Press, 2017.

Przybylla, M.: *From Embedded Systems to Physical Computing – Challenges of the “Digital World” in Secondary Computer Science Education*. Potsdam: Universität Potsdam, 2018 (Dissertation).  
<https://publishup.uni-potsdam.de/frontdoor/index/index/docId/41833>

RKI – Robert Koch Institut: *Epidemiologischer Steckbrief zu SARS-CoV-2 und COVID-19*. Berlin: Robert Koch Institut, 2021.  
<https://t1p.de/uxax>

Stahmer, V.: *CO2-Ampel – Erfahrungsbericht aus der Schule*. Hannover: heise online, 2020.  
<https://t1p.de/ep62>

UBA – Umweltbundesamt: *Das Risiko einer Übertragung von SARS-CoV-2 in Innenräumen lässt sich durch geeignete Lüftungsmaßnahmen reduzieren – Stellungnahme der Kommission Innenraumlufthygiene am Umweltbundesamt*. Dessau-Roßlau: Umweltbundesamt, 2020.  
<https://t1p.de/wkz6>

Umwelt-Campus Birkenfeld: *COVID-19 Prävention – CO2-Messung und bedarfsorientierte Lüftung*. Trier: Hochschule Trier, 2020.  
<https://t1p.de/brb8>

Vahid, F.; Givargis, T.D.: *Embedded System Design – A Unified Hardware/Software Introduction*. New York: Wiley, 2002.

Wörn, H.; Brinkschulte, U.: *Echtzeitsysteme*. Reihe „eXamen press“. Berlin; Heidelberg: Springer, 2005.

Wolf, W.: *Cyber-physical systems*. In: *Computer*, Vol. 42 (2009), No. 3, S.88–89.

## Weiterführende Internetquellen:

Datenblatt des verwendeten Sensors: *HW Electronics – TECHNICAL DATA MQ-135 GASSENSOR*.  
<https://t1p.de/3d1p>

*Physical-Computing-Material für das Schulfach „Digitale Welten“*:  
<https://github.com/maprzybylla/DW-PhysiComp>

Alle Internetquellen wurden zuletzt am 18. Februar 2021 geprüft und können auch aus dem Service-Bereich des LOG IN Verlags (<https://www.log-in-verlag.de/>) heruntergeladen werden.