SMB - Science Magic Box

<Autor> Luc Ivarra
<Autor> Marco Nicolini

<Info>

<Schlagwörter> Mikrocontroller, Wandler, Sensor, Aktor, Signal, physikalische Größe, Schleife, Verzweigung, sequenziell, Verarbeitung, Kalibrierung, Input, Output, lesen, schreiben, analog, digital, Linearität, Konversion, Steckplatine, Pin, Löten, Mensch-Maschine-Schnittstelle

<Unterrichtsfächer> Physik, Elektronik, Mathematik, IKT, Logik, Biologie

<Altersgruppe> 14–18 Jahre

<Hardware> Arduino UNO[1] mit Arduino DUE[2] und/oder TI-Nspire CX CAS mit TI-Innovator Hub

<Programmiersprache> C++ (Verwendung der Arduino IDE[3]) und/oder TI-Basic

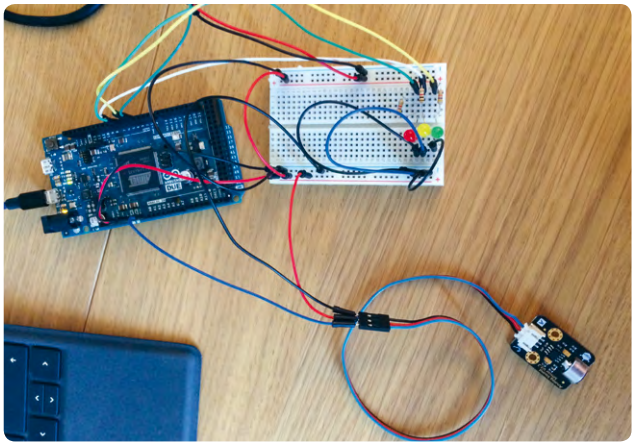
<Programmierniveau> mittel, mit einem Audioteil für fortgeschrittene Schülerinnen und Schüler

Eine Liste der verwendeten technischen Begriffe, Abkürzungen und Akronyme ist online verfügbar.[4]

<Zusammenfassung>

Die Schülerinnen und Schüler programmieren eine selbstkonstruierte Hardware-Software-Umgebung (basierend auf Arduino) und einen betriebsbereiten Taschencomputer (TI-Nspire CX CAS mit der Erweiterung TI-Innovator Hub). Beide Geräte werden für die Erfassung, Umwandlung und Übertragung von Sensordaten verwendet, sodass physikalische Größen leicht gehandhabt, gelesen, konvertiert und bedient werden können.

Die Schülerinnen und Schüler schreiben für Arduino ein Programm zu einer Anordnung mehrerer Sensoren, die physikalische Größen als Eingangssignale erfassen, und Aktoren, die auf die Datenerfassung reagieren. Letztere geben ein Ausgangssignal als physikalische Größe aus, nachdem ein Mikrocontroller das festgestellte Signal verarbeitet hat, sodass der korrekte Output festgelegt wird. (Siehe 1)



1: Arduino-Platine

TI Innovator Hub ist eine betriebsbereite Box, mit der die Schülerinnen und Schüler Programmiergrundlagen erlernen. Sie muss an den TI-Nspire CX CAS Taschenrechner angeschlossen werden. Die Box hat eine gute I/O-Schnittstelle, inklusive Helligkeitssensor, zwei LEDs und eingebautem Summer, der einen Ton in einer festgelegten Frequenz erzeugt. (Siehe 2)

2: TI-Nspire und TI-Innovator Hub

<Vorstellung des Konzepts>

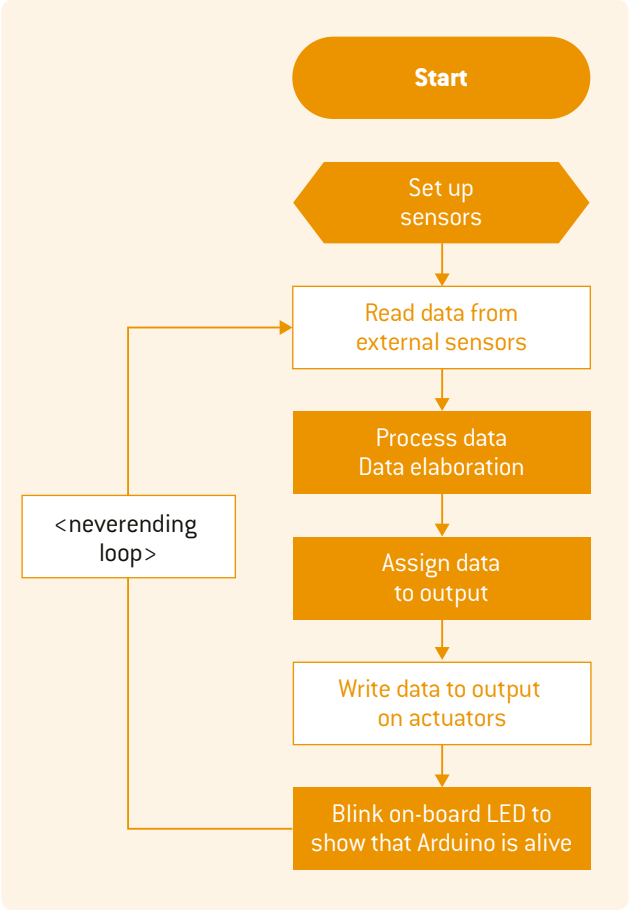
Die Einheit führt die Schülerinnen und Schüler in die Programmierung zu physikalischen Aufgabenstellungen ein, basierend auf der Messung physikalischer Größen, der Verarbeitung von Daten und der Entscheidung, welche Maßnahme unter Verwendung der Aktoren getroffen werden soll.

Das Programm besteht üblicherweise aus einer unendlichen Schleife (meist „lebt“ die Maschine, solange sie mit Energie versorgt wird, und sie muss die ganze Zeit arbeiten), in der die drei Aktionen Messen, Verarbeiten und Handeln in dieser Reihenfolge ausgeführt werden.

Die Schülerinnen und Schüler lernen, dass sie Programme schreiben können mit

1. sequenziellen Anweisungen
2. Schleifen (während … tue dies; wiederhole … bis)
3. Verzweigungen (wenn … dann … sonst),

wie aus dem Böhm-Jacopini-Theorem hervorgeht (siehe „Zusatzinformationen“[4]).



📷 3: Flussdiagramm

Das zweite Ziel der Einheit ist die Einführung in das Thema „Mikrocontroller". Die Schülerinnen und Schüler lernen, unter Verwendung von digitalen und analogen Ports, Sensoren und Output-Aktoren zu installieren und ein einfaches Programm zum Lesen des Inputs, Verarbeiten der Daten und Schreiben des Outputs zu erstellen. Als Output wählten wir entweder einen Ton oder Lichtsignale. Diese können als „Alarm" interpretiert werden, also als eine Warnung, die auf Basis eines von den Sensoren gelesenen Inputs ausgeben wird.

Die Schülerinnen und Schüler verwenden die integrierte Entwicklungsumgebung Arduino IDE[3], um in C++ zu programmieren. Es gibt hier betriebsbereite Funktionsbibliotheken, die das Programmieren erleichtern und beschleunigen.

Eine Steckplatine (siehe „Zusatzinformationen"[4]) muss vorhanden sein, damit die Schülerinnen und Schüler die Anordnung aufbauen und die Sensoren-Pins einfach an die Arduino I/Os, die 5V-Stromquelle und den GND-Pin anschließen können.

Die Struktur eines Programms ist in Metasprache in 📷 3 dargestellt.

Hinweis: Die Metaanweisung „*While (TRUE)*" („Während wahr") ist ein Trick, der jeden Prozessor dazu veranlasst, die darin enthaltenen Anweisungen unendlich oft (solange der Mikrocontroller über Energie verfügt) zu wiederholen.

Das dritte Ziel ist es, zu erlernen, wie eine erfasste physikalische Größe in eine andere umgewandelt (z. B. Lichtintensität in Schall) und nach außen übertragen werden kann. Dies geschieht wie folgt:

1. Das physikalische Signal (z. B. Licht, Schall, Kraft, Energie) wird durch den Sensor erfasst und in ein elektrisches Signal umgewandelt.
2. Das elektrische Signal wird in eine Zahl umgewandelt, die dem Prozessor zur Verfügung steht.
3. Die Zahl wird verarbeitet und vom Prozessor in eine andere Zahl umgewandelt und löst dann eine Aktion durch einen Aktorwandler aus.
4. Die Aktoren wandeln die Zahl in elektrische Signale um, die so als Output zur Verfügung stehen.
5. Das elektrische Signal wird schließlich in ein physikalisches Signal (z. B. Schall, Licht) umgewandelt.

In 1. und 5. müssen die Signale von einer Form in eine andere umgewandelt werden, wobei hier die Linearität der Umwandlung bzw. die „Nahezu-Linearität" extrem wichtig ist (siehe „Zusatzinformation"[4]).

Bezüglich der letzten Zeile der Metaprogrammierung („*Blink on-board LED*") siehe „Zusatzinformationen"[4] für eine detaillierte Erläuterung.

Üblicherweise wird das Inputsignal als „Stimulus" bezeichnet und kommt aus der Umgebung, in der die Sensoren zur Datenerfassung platziert wurden. Der Prozessor und das Programm sind so gestaltet, dass eine „Reaktion" mit mathematischen/logischen Operationen (von den Programmanweisungen durchgeführt und vom Mikrocontroller verarbeitet) auf den Stimulus erfolgt und dass diese „Antwort" als Output an die Umgebung abgegeben wird. In unserem Projekt ist die Umgebung der Raum um den Arduino, der dank Sensoren „sehen", „hören" und „Kräfte verspüren" kann.

Durch die Arbeit mit dem TI-Innovator Hub können sich die Schülerinnen und Schüler eher auf den Programmierteil konzentrieren, da die Mikrocontroller und Sensoren schon aufgebaut und betriebsbereit sind.

<Praktische Umsetzung>

Wir empfehlen mit einem Brainstorming zu beginnen, wobei die Vorstellungen der Schülerinnen und Schüler über Sensoren und automatische Maschinensteuerung diskutiert werden.

Diese Vorstellungen zu sammeln, praktische Erfahrungen zu machen und die Ergebnisse dann mit den ursprünglichen (vielleicht falschen) Annahmen zu vergleichen, hilft den Schülerinnen und Schülern das Ganze zu verstehen.

Dazu können Sie vorab einen Fragebogen vorbereiten:
↳ Weißt du, wie ein Thermostat die Raumtemperatur steuert?
↳ Wozu dient ein Einparksystem mit Tonsignalen? Wie reagiert der Fahrende, wenn das System einen Warnton abgibt?
↳ Habt ihr zu Hause einen Induktionsherd? Was signalisiert eine leuchtende LED?

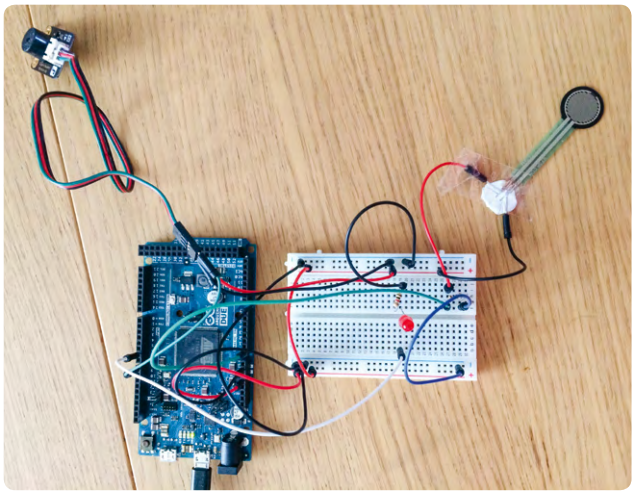
Siehe „PEC und Fragenliste“[4] für ein Beispiel einer Frageliste und die Verweise auf die „PEC“-Lehrmethode (Prevision, Experience, Correction).

<Theoretische Phase mit Arduino[1]>

Die Lehrkraft führt die Schülerinnen und Schüler in die C++-Programmierung[3] mit Struktur und Basisanweisungen ein, sodass sie eine einfache Schleife mit den Anweisungen *analogRead, digitalRead, analogWrite, digitalWrite, if…then…else, loop, while* schreiben können.

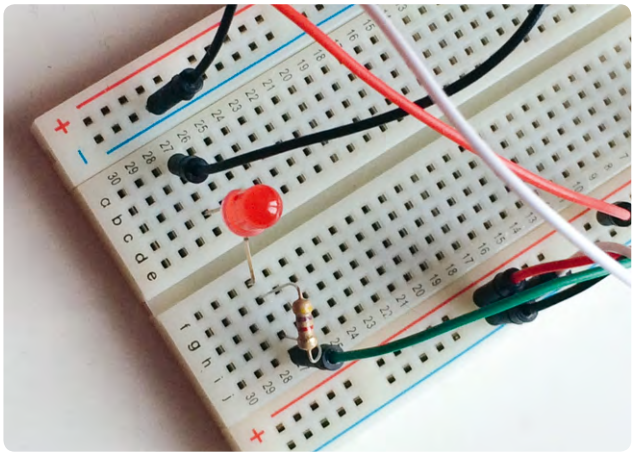
Hardware-Teil

Die Lehrkraft stellt die Anordnung des Mikrocontrollers vor und zeigt dabei den Mikroprozessor, die analogen I/O-Pins (Verbindungen) und die digitalen Input-/Output-Pins (Verbindungen).



📷 4: Arduino, Steckplatine und Sensoren

Die Schülerinnen und Schüler lernen, dass jeder Sensor/Aktor meist mehrere Verbindungen aufweist:
↳ zum 5V- oder 3,3V-Arduino-Output für die Stromzufuhr
↳ zum GND-Signal (Erdung), damit der Strom fließt
und
↳ zu einem weiteren digitalen oder analogen Input-Pin, wenn Daten von außen gelesen (erfasst) werden
oder
↳ zu einem weiteren digitalen oder analogen Output-Pin, wenn eine Aktion nach außen erfolgen soll, z. B. Licht oder einen Ton ausgeben oder etwas Anderes, das eine Situation anzeigt (eine „Write“-Operation)



📷 5: Steckplatine im Detail

Software-Teil

Ein einfaches Programm, das einen Sensor liest und einen Aktor schreibt, wird von der Lehrkraft vorgestellt. Dabei sollen die Schülerinnen und Schüler eine klare Assoziation zwischen physischen Pins und physischer Onboard-Adresse des Mikrocontrollers herstellen können.

Ein Beispiel für betriebsbereite Anweisungen ist online verfügbar („Programmbeispiel 1“[4]).

Die Schülerinnen und Schüler müssen dabei berücksichtigen, dass der Prozessor die Programmanweisungen eine nach der anderen und in der geschriebenen Reihenfolge ausführt. Nur die Schleife weicht von diesem Grundsatz ab, da sie den Prozessor veranlasst, die in der Schleife enthaltenen Anweisungen zu wiederholen, solange der Mikrocontroller über Strom verfügt.

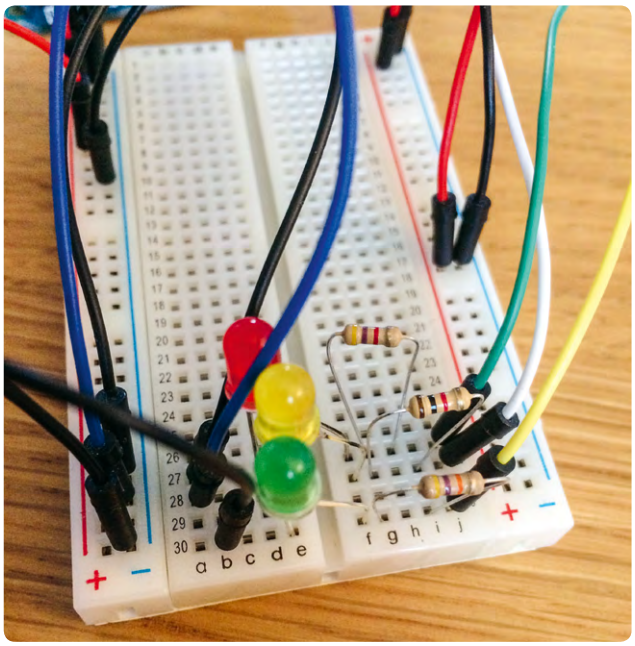
<Praktische Phase mit Arduino>

Die Schülerinnen und Schüler machen praktische Erfahrungen mit dem Mikrocontroller, der Steckplatine und den Sensoren. Die Lehrkraft sollte die Struktur der Steckplatine vorstellen und dabei alle verfügbaren Verbindungen aufzeigen und erläutern, wie die 5V-, die GND- und die I/O-Signale vom Arduino[1] zur Steckplatine gebracht werden können. Die Schülerinnen und Schüler werden dann aufgefordert, das Programmierbeispiel zu kopieren und das Programm mit den verbundenen I/Os auszuprobieren, zu testen und wenn nötig zu korrigieren („Debugging“).

Hardware-Teil

Die Schülerinnen und Schüler benötigen den Mikrocontroller, die Sensoren und kurze Kabel (10 cm), um Verbindungen zwischen den Sensoren-Pins und den Federkontakten der

Steckplatine herzustellen. Manchmal ist es notwendig, zusätzliche Kabel an die Sensoren zu löten.



6: Steckplatine mit LEDs

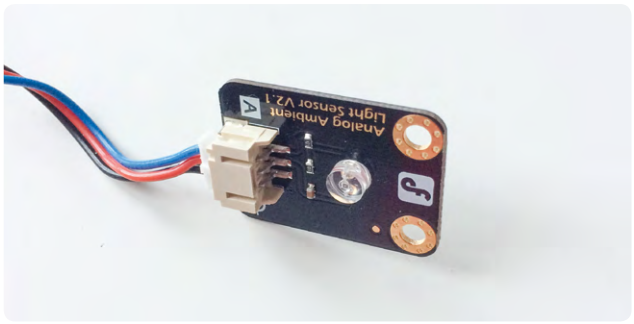
Mithilfe der kurzen Kabel werden die 5V-Stromquelle und das GND-Signal an die Steckplatine angeschlossen sowie die analogen/digitalen Pins des Arduino[1] an einige Federkontakte der Steckplatine. Dies ermöglicht es, die Sensoren an der Steckplatine zu befestigen und die erforderlichen elektrischen Signale zu erhalten. (Siehe 6)

Software-Teil

Zuerst überprüfen die Schülerinnen und Schüler, ob die Verbindungen zwischen Mikrocontroller und Steckplatine korrekt funktionieren, indem sie auf die genaue Übereinstimmung zwischen der logischen Nummer des Pins am Mikrocontroller und dem Sensor-Pin auf der Steckplatine achten. Anschließend versuchen sie, ein einfaches Programm zu schreiben, welches online verfügbar ist („Programmierbeispiel 1“[4]).

Algorithmen mit Arduino

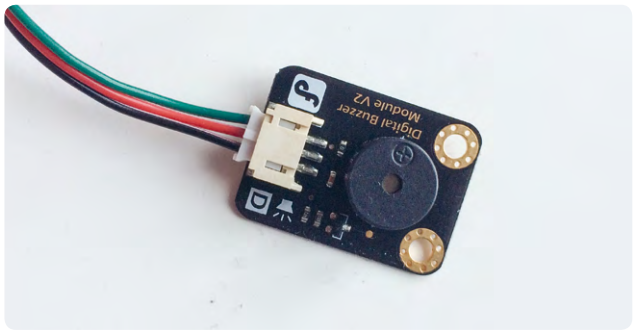
Wir haben verschiedene Signalumwandlungen von einer physikalischen Form in eine andere vorbereitet.



7: Lichtsensor

Umwandlung eines analogen Lichtsignals in ein digitales Lichtsignal in einer LED (moduliert mit einem PWM-Feature) und in einen Ton: Die ausgegebene Tonfrequenz erhöht sich mit der Lichtintensität. Praktische Anwendung: Wecker, Hilfe für Sehbehinderte. (Siehe 7)

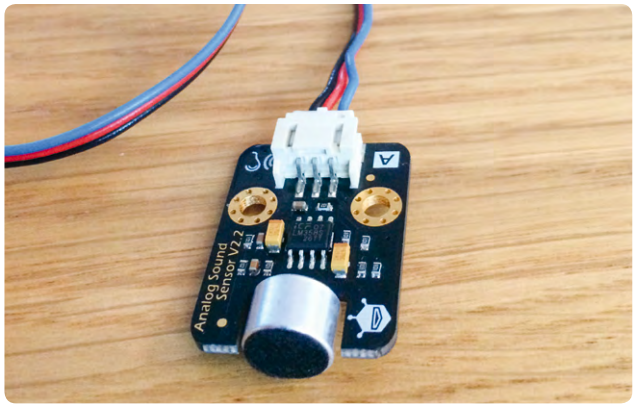
Umwandlung einer durch den Kraftsensor festgestellten Kraft in ein digitales Lichtsignal in einer LED (moduliert mit einem PWM-Feature) und in einen Ton: Die Lichtintensität erhöht sich mit der Kraft. Praktische Anwendung: Überlastungsalarm (Siehe 8)



8: Digitaler Summer

Umwandlung eines externen Geräuschsignals in ein digitales Lichtsignal in einer LED. Je höher der Ton, desto höher ist die Lichtfrequenz der eingeschalteten LED. Praktische Anwendung: Lärmbelastungskontrolle.

Umwandlung einer von einem Distanzsensor gemessenen Distanz in einen Ton. Technische Anwendung: Einparksensoren. (Siehe 9)



9: Distanzsensor

Umwandlung eines Temperatursignals in einen Ton und ein Lichtsignal. Technische Anwendung: Ofentemperaturkontrolle.

Umwandlung einer Bodenfeuchtigkeitskonzentration in ein Lichtsignal. Technische Anwendung: Pflanzenbewässerungs- und Gieß-Alarm und -Kontrolle. (Siehe 10)

10: Bodenfeuchtigkeitssensor

Bezüglich des Programms für diese Anwendungen auf der Arduino-Platine siehe „Programmierbeispiel 2“[4].

Alle diese Algorithmen sind Signalkontroll- und Warnsysteme, die eine gewählte Umgebung bezüglich einer physikalischen Größe überwachen und eine Warnung abhängig vom Input (Stimulus) ausgeben.

<Theoretische Phase mit TI-Innovator Hub>

Hardware-Teil

Die Lehrkraft kann demonstrieren, wie intuitiv und einfach es ist, die Sensoren zu verbinden.

Software-Teil

Für die Basisprogrammierung wird nur der Taschenrechner benötigt. Die Schülerinnen und Schüler müssen lediglich die oben erwähnten TI-Basic-Anweisungen beherrschen. Dann kann der Hub verbunden werden, und sie lernen, wie sie damit kommunizieren und insbesondere wie sie die Anweisungen „Read“ und „Get“ für die Datenerfassung und „Set“ für die Kontrolle der Outputs einsetzen.

<Praktische Phase mit TI-Innovator Hub>

Hardware- und Software-Teil

Die Schülerinnen und Schüler beginnen mit Beispielen zu den Grundlagen, um sich mit dem Hub vertraut zu machen, bevor sie sich an offenere Aufgabenstellungen wagen. Anhand kleiner Übungen eignen sie sich an, wie sie die verschiedenen Outputs steuern, z. B. Steuerung der LED-Farbe, LED blinken lassen, Dauer des Blinkens steuern und Töne mit einer gewissen Frequenz erklingen lassen. Die unendliche Schleife ist auch hier die Basisstruktur, um Operationen unendlich ausführen zu lassen.

Algorithmen mit TI-Innovator Hub

Die Schülerinnen und Schüler lösten zwei Aufgaben: Programmierung eines automatischen Schalters, der die Beleuchtung erst dann einschaltet, wenn die Intensität des Umgebungslichts unterhalb einer gewissen Schwelle fällt, und Programmierung eines Weckers, der einen Ton in einer Frequenz ausgibt, die mit der Erhöhung der Umgebungslichtintensität ansteigt. Weitere Entwicklungen sind möglich, wozu jedoch zusätzliche Sensoren gekauft und mit dem Hub verbunden werden müssten.

<Kauf der Sensoren>

Hinweise zum Kauf der Sensoren sind online verfügbar.[4]

<Fazit>

Am Ende dieses Projekts bemerkten wir bei unseren Schülerinnen und Schülern eine deutliche Verbesserung des Verständnisses von Programmierung und allgemeiner Programmstruktur sowie von Logik und Algorithmen.

<Kooperationsmöglichkeiten>

Eine wundervolle Kooperationsmöglichkeit wäre ein „Selbstunternehmer“-Projekt, bei dem die Schülerinnen und Schüler versuchen könnten, eine eigene Mensch-Maschine-Schnittstelle (human machine interface, HMI) mit einer gewissen technischen Nützlichkeit zu erfinden. Diese HMI sollte einen Stimulus der Umgebung lesen (Atmosphäre, Wohnung, menschlicher Körper usw.) und reagieren, indem sie ein weiteres Signal ausgibt, das eine Warnung ertönen lässt, eine Aktion durchführt oder eine Situation anzeigt. Partnerschulen im Ausland können eine Marktumfrage durchführen, um zu verstehen, wie die Marktnachfrage und der Marktwert sind, die das Gerät in ihren Ländern haben könnte. Jede Schule, die an einem solchen Austausch teilnimmt, entwickelt ein Gerät und führt Marktumfragen für ein Produkt durch, das von den anderen Schulen entwickelt wird. Am Ende des Projekts könnte das vom Markt am meisten nachgefragte Gerät von einem Partnerunternehmen im kleinen Rahmen produziert und verkauft werden. Auf der ganzen Welt hat Selbstunternehmertum einen hohen Stellenwert als Chance, Themen aus Wissenschaft, Technologie und Wirtschaft im Zusammenhang verständlich zu machen.

<Quellen und Hinweise>

[1] www.arduino.cc
[2] www.arduino.cc/en/Guide/ArduinoDue
[3] www.arduino.cc/en/Main/Software
[4] Sämtliches Zusatzmaterial ist erhältlich auf www.science-on-stage.de/coding-materialien.

<Impressum>

<Entnommen aus>
Coding im MINT-Unterricht
www.science-on-stage.de/coding

<Herausgeber>
Science on Stage Deutschland e.V.
Am Borsigturm 15
13507 Berlin

<Revision und Übersetzung>
Translation-Probst AG

<Gestaltung>
WEBERSUPIRAN.berlin

<Illustration>
Rupert Tacke, Tricom Kommunikation und Verlag GmbH

<Text- und Bildnachweise>
Die Autorinnen und Autoren haben die Bildrechte für die Verwendung in dieser Publikation nach bestem Wissen geprüft und sind für den Inhalt ihrer Texte verantwortlich.

<Bestellungen>
www.science-on-stage.de
info@science-on-stage.de

<ISBN PDF-Fassung>
978-3-942524-60-5

1. Auflage 2019

Science on Stage Deutschland -
The European Network for Science Teachers

… ist ein Netzwerk von Lehrkräften für Lehrkräfte aller Schularten, die Mathematik, Informatik, Naturwissenschaften und Technik (MINT) unterrichten.
… bietet eine Plattform für den europaweiten Austausch anregender Ideen und Konzepte für den Unterricht.
… sorgt dafür, dass MINT im schulischen und öffentlichen Rampenlicht steht.

Science on Stage Deutschland e.V. wird maßgeblich gefördert von think ING., der Initiative für den Ingenieurnachwuchs des Arbeitgeberverbandes GESAMTMETALL.

Machen Sie mit!
www.science-on-stage.de
www.facebook.com/scienceonstagedeutschland
www.twitter.com/SonS_D

Bleiben Sie informiert!
www.science-on-stage.de/newsletter

Ein Projekt von
SCIENCE ON STAGE DEUTSCHLAND

Hauptförderer von Science on Stage Deutschland
think ING.
Die Initiative für Ingenieurnachwuchs

Mit freundlicher Unterstützung von
SAP