

Unterstützende Werkzeuge zum Einstieg in die Algorithmik (Ozobots, Calliope Mini, Arduino)

Kiesmüller, U.
Simon-Marius-Gymnasium Gunzenhausen

DOI: 10.18420/ibis-01-01-08

Zusammenfassung

In diesem Artikel werden Einsatzmöglichkeiten verschiedener Werkzeuge im Informatikunterricht der Sekundarstufe für den Themenbereich Algorithmik bzw. Einstieg ins Programmieren vorgestellt. Der Fokus wird hierbei auf die Mikrocontroller-Boards bzw. Roboter Calliope Mini, Ozobot und Arduino gelegt. Den Ausführungen liegen jahrelange Erfahrungen zugrunde, die beim Einsatz dieser Werkzeuge im Informatik-Unterricht der 7. Jahrgangsstufe (in Bayern) gesammelt wurden. Da der Ozobot ab Beginn des Schuljahres 2023/2024 auch in der neu strukturierten 11. Jahrgangsstufe des neuen bayerischen Gymnasiums im Bereich Algorithmik eingesetzt werden wird, ist dessen Betrachtung besonders ausführlich gestaltet.

Learning by doing

*„Was du mir sagst, das vergesse ich.
Was du mir zeigst, daran erinnere ich mich.
Was du mich tun lässt, das verstehe ich.“*

Dieses Zitat des chinesischen Philosophen Konfuzius (551 - 479 v. Chr.) ist sicher allseits bekannt. Nichtsdestotrotz findet es sich auch immer wieder in modernen fachdidaktischen Aussagen. Und wer hat noch nicht in irgendeinem Zusammenhang „*learning by doing*“ gehört?

Beim Einstieg in die Welt des Programmierens in der Sekundarstufe ist dies sicherlich unstrittig. Ein weiterer Aspekt zur Verstärkung der Motivation, sich mit diesem eigentlich abstrakten Thema zu befassen und dabei möglichst schnell sichtbare Erfolge zu erzielen, ist die Auswahl geeigneter Werkzeuge. Um auch haptische Effekte für selbstständiges Entdecken, leichtere Einprägsamkeit und tiefergehendes Verständnis nutzen zu können, ist die Verwendung von Mikrocontroller-Boards und/oder Minirobotern in Erwägung zu ziehen. Bevor Einsatzmöglichkeiten von Calliope Mini, Ozobot und Arduino Uno gegenübergestellt werden, werden im nächsten Abschnitt zunächst einige generelle Aspekte zum Programmierenlernen

in der Sekundarstufe sowie ein möglicher Unterrichtsvorlauf für die Arbeit mit den genannten Werkzeugen vorgestellt.

Programmierenlernen in der Sekundarstufe

In vielen Lehrplänen finden sich ähnliche Punkte wie sie im bayerischen Lehrplan für das neue bayerische Gymnasium mit neunjähriger Ausbildungsdauer aufgeführt sind:

Die Schülerinnen und Schüler setzen unter sinnvoller Nutzung algorithmischer Bausteine einfache Algorithmen mithilfe geeigneter Programmierwerkzeuge um (ISB: LehrplanPLUS – Natur und Technik 7 – Informatik).

Fachbegriffe: Algorithmus, Anweisung, Sequenz, ein- und zweiseitig bedingte Anweisung, Wiederholung mit fester Anzahl, Wiederholung mit Bedingung (ISB: LehrplanPLUS – Natur und Technik 7 – Informatik)

Alle weiteren Betrachtungen basieren auf Erfahrungen im Informatik-Unterricht der 7. Jahrgangsstufe im Rahmen des Faches Natur und Technik.

Umgesetzt werden können die oben aufgeführten Lehrplananforderungen mit Softwareprogrammen, die (auch) als Browseranwendungen zur Verfügung stehen. Eine der bekanntesten hiervon ist Scratch (<https://scratch.mit.edu>), eine erstmals 2007 veröffentlichte bildungsorientierte visuelle Programmiersprache für Kinder und Jugendliche inklusive

Entwicklungsumgebung und einer eng verknüpften Online-Community-Plattform.

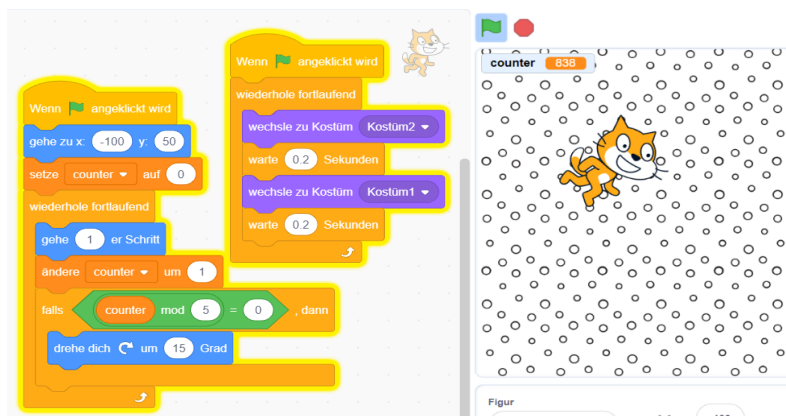


Abbildung 1: links - Skript bestehend aus Programmierblöcken; rechts - Bühne mit Figur zur Programmausführung

Scratch bietet die Möglichkeit der blockbasierten Programmierung per drag&drop. Da die Programmieroberfläche durch Farbgebung und Form die Funktionsweise der einzelnen Bausteine schnell wiedererkennbar macht, müssen die Lernenden nur wenig lesen und durch die Art der blockbasierten Programmierung auch nahezu nichts

schreiben. Dies sollte beim Lehren von Programmierung unbedingt berücksichtigt werden, um Lernende mit Problemen beim Lesen und Schreiben nicht auszugrenzen und Einstiege bereits für jüngere Altersgruppen gestalten zu können.

Selbst gestaltbare Figuren und Hintergründe bieten viele kreative Freiräume für die Lernenden. Außerdem können auch eigene Programmierbausteine definiert werden.

Divers diskutieren lässt sich der Gesichtspunkt der parallelen Ausführung mehrerer Programmteile bzw. Programme. Da die Lernenden sich diese beim intuitiven Vorgehen selbst erschließen können (und sehr gerne einsetzen), ist sie der kreativen Gestaltung eigener Programme sicher zuträglich. Ob aber ein tatsächliches Verständnis für paralleles Programmieren entsteht, ist bei sich in der Art der Aktivitäten in den Skripten überschneidenden oder mit einer großen Zahl von Programmteilen versehenen Projekten eher in Frage zu stellen.

Selbst definierbare Variablen, Text- und Sound-Ausgaben bieten ein reichhaltiges Repertoire für sehr unterschiedliche Aufgabenstellungen durch die Lehrkraft und insbesondere selbstständiges, kreatives Arbeiten der Lernenden.

Alle genannten Vorzüge besitzen auch die im nächsten Abschnitt vorgestellten Werkzeuge (mit ihren jeweiligen Programmiermöglichkeiten), die den zusätzlichen Vorteil bieten, dass die Lernenden etwas in der Hand haben, im wahrsten Sinne des Wortes „begreifen“ können und nicht „nur“ den Bildschirminhalt betrachten. Die hier betrachtete Geräteauswahl wurde auch im Hinblick darauf getroffen, dass die verwendeten Geräte klein genug sind, damit sie auch auf einem „vollen“ Arbeitsplatz mit Computer, Monitor, Schnellhefter und Federmäppchen gut eingesetzt werden können.

Mikrocontroller-Boards und Roboter

Beim Einsatz zusätzlicher Geräte hat sich in den vergangenen Jahren in den Unterrichtsstunden des Autors folgendes Vorgehen bewährt.

Die Einstiegsstunde zur Informatik wird so gestaltet, dass von den Lernenden gesammelt wird, was für sie alles zur *Informatik* gehört. Hierbei wurde jedes Mal auch mindestens ein Aspekt genannt, der zum Bereich *Robotik* führte. Die Stunde schloss mit der Stellung der Hausaufgabe einen Roboter zu zeichnen (Funktionsweise sollte im Bild erkennbar sein) – das Bild ist dann jeweils das Titelblatt des eigenen Informatikschnellhefters. In der Anschlussstunde wurden gemeinsam mehrere Bilder vergleichend betrachtet und herausgearbeitet, was einen Roboter ausmacht. Basierend auf den Aussagen der Lernenden wurden – in Teilen angelehnt an Formulierungen in Lehrbüchern – folgende Aspekte zusammengestellt:

Ein Roboter ist

- eine technische Apparatur, die dazu dient, Menschen bei häufig wiederkehrenden (mechanischen) Arbeiten zu unterstützen.
- Hierbei haben sie über Sensoren (z. B. für Wärme, (Ultra-)Schall, UV-Licht) Kontakt mit ihrer Umwelt.
- Aktoren (z. B. Motoren, Displays) dienen zur Bewältigung der Aufgaben.
- Sie werden gesteuert von Algorithmen.

(möglicher Hefteintrag)

Somit ist der Übergang zur Algorithmik bereits geschafft und auch hier kann eine Begriffsfestlegung mit den Lernenden erarbeitet werden. Wiederum basierend auf ihren Aussagen und angelehnt an die Formulierungen in Lehrplanmaterialien wurden folgende Aspekte (aus fachwissenschaftlicher Sicht evtl. nicht ganz vollständig) zusammengestellt:

Ein Algorithmus ist eine

- endliche Folge von
- eindeutigen Anweisungen, die bei
- einheitlichen Voraussetzungen zu einem
- eindeutigen Ergebnis führen.

(möglicher Hefteintrag)

Nach diesem kurzen Theorieteil, der für alle in den nächsten Abschnitten betrachteten Werkzeuge identisch ist, kann nun ohne Verzögerungen in die Praxis gestartet werden.

Calliope Mini

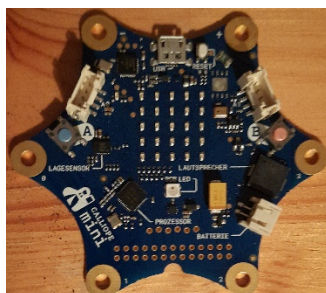


Abbildung 2: Calliope mini
(erste Generation)

Calliope mini ist ein Microcontroller-Board mit diversen bereits installierten Ein- und Ausgabemöglichkeiten, das für Bildungszwecke entwickelt wurde. Es ist insbesondere für den Einsatz mit jüngeren Schülerinnen und Schülern konzipiert, eignet sich aber – speziell durch mögliche Erweiterungen mit externen Sensoren und einem MotionKit – auch sehr gut für den Einsatz in der Sekundarstufe. Den Lernenden wird ein spielerischer Einstieg ins Programmieren und das algorithmische Denken ermöglicht.

Für die blockbasierte Programmierung per drag&drop kommt (zum Beispiel) die Onlineanwendung MakeCode (<https://makecode.calliope.cc>) zum Einsatz. Diese bietet auch eine Vorschau, die den Programmierenden jeweils sofort anzeigt, wie sich Änderungen im

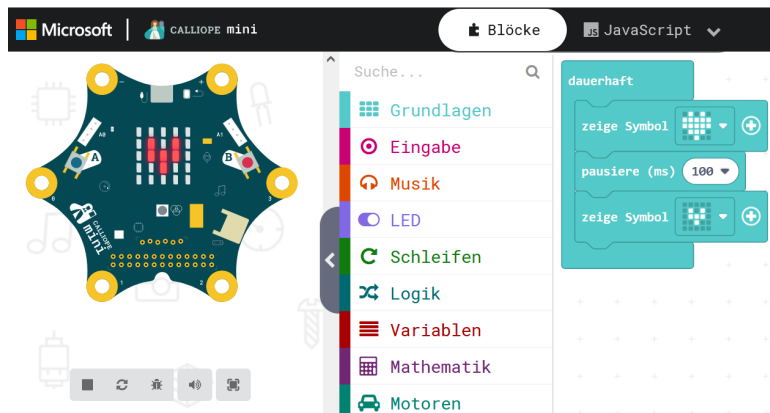


Abbildung 3: MakeCode mit Vorschau zur Blockly-Programmierung

Programmcode auf den Programmablauf im Microcontroller-Board auswirken. Das Calliope mini verfügt über verschiedene Eingabemöglichkeiten: z. B. verschiedene Tasten, mehrere Kontakte oder ein Bewegungssensor können eingesetzt werden. In der Vorschau werden für alle nicht direkt erreichbaren Eingabemöglichkeiten jeweils zusätzliche Buttons mit eingeblendet (z. B. „Schütteln“, „A & B“). Als

Ausgabemöglichkeiten bietet Calliope mini Zahl-, Text- und Symbolausgabe im LED-Feld, einzelne LED und Sound. Zur Standardausrüstung des Calliope mini gehört auch eine Batteriebox zur Stromversorgung ohne USB-Anschluss.

Bereits in der Grundausrüstung lassen sich mit den Lernenden attraktive Programme gestalten. Programmierbausteine für die algorithmischen Grundstrukturen sowie die einfache Umsetzung eines Variablenkonzepts sind vorhanden.

Zur Einführung der Funktionsweise sind Standardprogramme als Voreinstellung vorhanden. Eine erste Programmieridee kann sein, eine startende Rakete zu gestalten – das lässt sich mit einer einfachen Sequenz erreichen, welche die sukzessive Symbolausgabe auf dem LED-Feld steuert. Eine dazu passende Soundausgabe macht den Lernenden viel Freude. Sehr schnell lässt sich durch das Vorschalten eines Countdowns den Programmieranfängern die Notwendigkeit des Einsatzes von Variablen sowie die Verwendung von Wiederholungen nahebringen.



Video zur Aufgabe „Raketenstart“

Zweite Programmidee kann z. B. ein „Würfel für Blinde“ sein – Calliope mini soll beim Schütteln eine zufällige Zahl von 1 bis 6 anzeigen und deren Wert auch durch eine passende Soundausgabe wiedergeben. Hierbei benötigen die Programmierenden dann zusätzlich bedingte Anweisungen, so dass die algorithmischen Grundstrukturen bereits an dieser Stelle komplett in Programmen eingesetzt werden. Der beim „Blindenwürfel“ bereits erscheinende „soziale Gedanke“ kann dann auch bei Aufgabenstellungen für komplexere Aufgaben verwendet werden: „Gestalte einen Rechentrainer für z. B. jüngere Geschwister. Calliope erstellt

aus zwei zufällig gewählten Ziffern eine einfache Rechenaufgabe (Addition oder Multiplikation). Mit den Tasten A und B kann das Ergebnis eingegeben werden (z. B. durch einfaches Hinaufzählen mit der Taste A, während B zum Hinunterzählen dient oder durch Hinaufzählen der Zehnerstelle mit Taste A und Hinaufzählen der Einerstelle mit Taste B). Calliope soll das Ergebnis automatisch prüfen und bei korrektem sowie bei falschem Ergebnis jeweils eine passende Ausgabe erzeugen.“ Um die Ergebniseingabe jeweils abzuschließen kann hierbei das gleichzeitige Drücken der Tasten A und B verwendet werden. Durch das „Prellen“ der Tasten kann es dabei zu Fehleingaben kommen, selbst wenn das Programm korrekt gestaltet ist (prüfbar mit dem zusätzlichen Button „A & B“ in der Vorschau von MakeCode). Dieses rein technische Problem lässt sich aber auf verschiedene Weise lösen – z. B. können an geeigneter Stelle im Programm kurze Verzögerungen mit dem „Warte“-Baustein eingefügt werden. Diese Aufgabenstellung beschäftigt die Lernenden über mehrere Unterrichtsstunden hinweg. Durch die reichlichen Binnendifferenzierungsmöglichkeiten (erst nur Additionstrainer, dann einen für Multiplikation, abschließend einen Rechentrainer, der Addition bzw. Multiplikation zufällig wählt oder ein Subtraktionstrainer) kommt bei den Lernenden auch keine Langeweile auf. Auch Wartezeiten bei Fragestellungen an die Lehrkraft können die Programmierenden immer z. B. dadurch überbrücken, dass sie ihre Bild- und Tonausgabe für „korrekt“ bzw. „falsch“ weiter ausgestalten.



Video zur Aufgabe „Multiplikationstrainer“

Die Übertragung des als HEX-Datei speicherbaren Programms kann mittels einer im Standardzubehör vorhandenen USB-Verbindung erfolgen. Bei Calliope-Geräten der neueren Generationen ist zusätzlich eine Programmübertragung mit Hilfe einer App von einem mobilen Endgerät per Bluetooth möglich.

Sollte man den Lernenden den Programmcode nicht nur in Blockly-Darstellung (oder den HEX-Code in der zugehörigen Textdatei) zeigen wollen, so bietet MakeCode eine Übersetzungsmöglichkeit in Python- oder Java-Script. Somit kann bei Bedarf auf die gesamte Spannweite von blockbasierter Programmierung über eine Hochsprache zum Programmieren bis zum maschinennahen Code eingegangen werden.

Durch den Einsatz der frei belegbaren – gut mit Krokodilklemmen nutzbaren – Pins und als Zusatzzubehör erhältlichen Feuchtigkeits-, Ultraschall-Entfernungs- oder CO₂-Sensoren wird die Ideenvielfalt der Lernenden für Programme noch erweitert. Mit Grove-Konnektoren, für die es nahezu alles an Sensoren und Aktoren gibt, kommen wiederum sehr viele Möglichkeiten hinzu. Steht viel Zeit zur Verfügung, können durch die Anschaffung eines MotionKits, mit dem Calliope dann zum „echten Roboter“ wird, die Einsatzideen nochmals umfassender gestaltet werden.

Ein Gerät, das bereits als Roboter gestaltet ist und einige Sensoren standardmäßig eingebaut hat, wird im nächsten Abschnitt ausführlich betrachtet.

Ozobot



Abbildung 4: Ozobot evo mit Abstandssensoren an der Rückseite

Ozobots sind für Kinder und Jugendliche geeignete, programmierbare Roboter, die für einen Einsatz in teilweise beengten Platzverhältnissen in Unterrichtsräumen gut geeignet sind. Der Ozobot ist ein kleiner Roboter mit eingebautem Akku und fünf Farb-Sensoren auf der Unterseite.

Es gibt ihn in zwei Varianten, den *Ozobot bit* und den *Ozobot evo*. Der *Ozobot bit* lässt sich ganz ohne Computer über Abfolgen von Farben und Linienverläufe steuern. Der *Ozobot evo* verfügt über deutlich mehr Sensoren. Diese sowie seine Aktoren sind alle einzeln in den von den Lernenden selbst erstellten Programmen ansteuerbar. Somit ist er auch für die Umsetzung komplexerer Programmideen geeignet.

Eine detaillierte Betrachtung der beiden Varianten des Ozobot folgt in den nächsten Abschnitten.

Ozobot bit

Er besitzt optische Zeilen- und Farbsensoren, seine Aktoren umfassen die Motoren der Räder sowie eine programmierbare LED. Sein Akku kann über USB-Kabel geladen werden – hierbei ist ca. eine Stunde Ladezeit für vier Stunden Laufzeit notwendig, sodass er sehr gut für den Einsatz an einem Unterrichtsvormittag geeignet ist. Zur Sicherheit sollte man aber eine Ladepause einplanen bzw. Ersatzgeräte parat haben, wenn ein Akku „leergefahren“ ist.

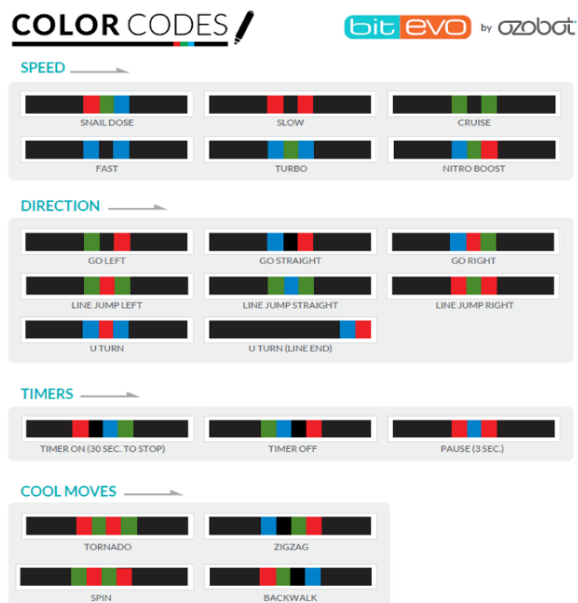


Abbildung 5: Ozobot-Farbcodes (Ausschnitt)
(Ozobot (2023))

Ozobot bit verfügt über einen fest programmierten „Linienfolger“ und sein Verhalten während der Fahrt kann über Farbfolgen, welche auf der schwarzen Linie, der er folgt, eingetragen sind, beeinflusst werden. Geschwindigkeit, Richtung (bei Abzweigungen) und Art der Bewegung sind steuerbar – er kann auch von einem Linienstück zu einem anderen „springen“. Außerdem zeigt die LED die Farbe der durchgehenden Linie an, auf der er sich gerade befindet. Bei zusätzlicher Verwendung der Codes für eine Pause von drei Sekunden und für Start bzw. Stopp eines 30-Sekunden-Timers lassen sich kreative Wettrenn- und Rätselaufgabenstellungen entwickeln.

Die Steuerung über Farbcodes lässt sich in das Unterrichtsthema „Algorithmik“ im Teilbereich

„bedingte Anweisungen“ (z. B. „falls rot-grün-blau fahre im Schneckentempo weiter“) oder „Wiederholungen“ (bei manchen Farbcodes mit Zähler, bei anderen mit Bedingung) integrieren.

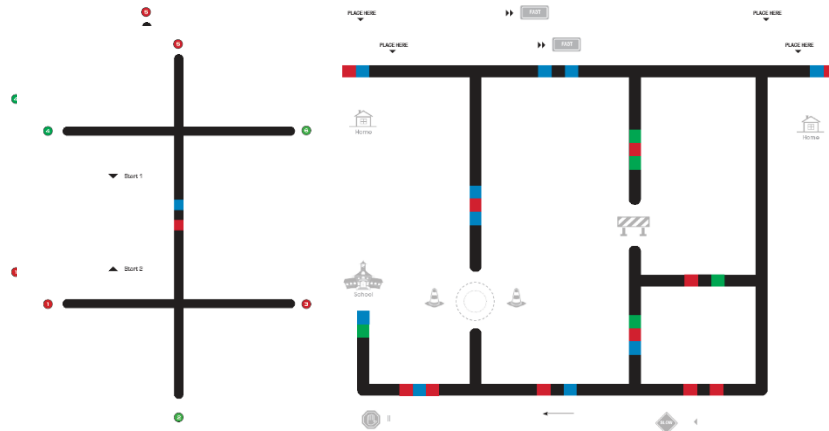


Abbildung 6: Parcours zur Erforschung der Farbcodes
Geier, Gerald (2017); Geier, G., Ebner, M. (2017)

Insbesondere auch zum Kennenlernen grundlegender Funktionsweisen von Ozobots lassen sich die Ozobot bit wunderbar einsetzen. Die Schülerinnen und Schüler untersuchen die Fahrt auf vorgefertigten Karten und stellen erste Vermutungen über die Bedeutung bestimmter Farbcodes auf. Durch Austesten von Farbcodes mit selbstgemalten Pfaden können sie ihre Vermutungen überprüfen und Schritt für Schritt eine selbsterstellte Farbcode-Tabelle bis (nahezu) zur Vollständigkeit ergänzen. Daran anschließend kann mit den im nächsten Abschnitt näher betrachteten Ozobot evo tiefer in die Algorithmik und das Erlernen der Programmierung eingestiegen werden.

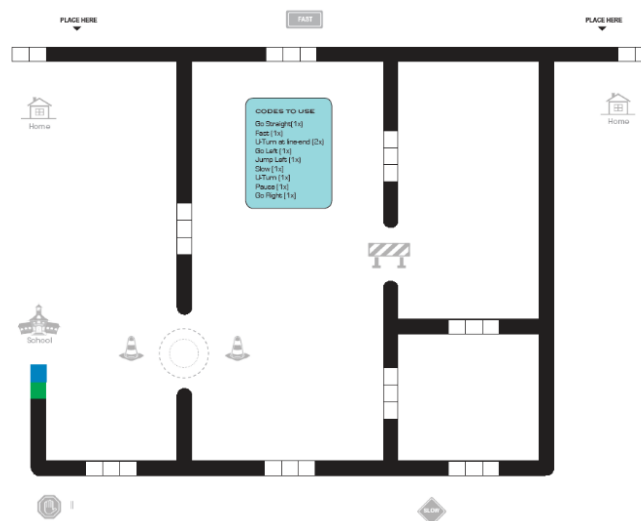


Abbildung 7: Aufgabe: Korrektes Einsetzen bekannter Farbcodes
Geier, Gerald (2017); Geier, G., Ebner, M. (2017)

Ozobot evo

Zusätzlich zu den optischen Zeilen- und Farbsensoren besitzt er sowohl an der Frontseite links und rechts als auch an der Rückseite Infrarot-Abstandssensoren, deren Rückmeldungen in Programmen jeweils abfragbar sind. Seine Aktoren umfassen die einzeln ansteuerbaren Motoren der Räder sowie sechs programmierbare LEDs und einen Lautsprecher zur steuerbaren Soundausgabe. Insbesondere wegen der vielfachen Licht- und Soundeffekte ist eine Ladezeit von ca. einer Stunde nur noch für ungefähr eine Stunde Laufzeit ausreichend. Berücksichtigt man aber, dass während des Programmierens die Ozobots nicht in Betrieb sein müssen, ist bei guter Planung mit einem Klassensatz der Einsatz in einer Doppelstunde sehr gut möglich. Somit kann man ihn mit einer Ladepause durchaus in vier von sechs Vormittagsstunden einsetzen, ohne dass ein Akku „leergefahren“ wird. *Ozobot evo* kann über die Online-Anwendung *OzoBlockly* oder eine im App Store bzw. bei Google Play kostenfrei erhältliche *EvoApp* programmiert werden. Zur Nutzung der App sollte man ein Tablet oder Ähnliches als mobiles Endgerät einsetzen, da die App nicht für den Einsatz auf kleinen Handybildschirmen gestaltet ist. Um lediglich selbst erstellte Programme auf einen *Ozobot evo* zu übertragen, ist die App aber auch auf einem Handy sehr gut einsetzbar. Grundsätzlich erfolgt die Programmübertragung von *OzoBlockly* zum *Ozobot* durch „flashing“. Der Programmcode wird in eine Farbabfolge übersetzt, die dann auf dem Bildschirm flackernd erscheint. Der *Ozobot evo* muss dabei an der entsprechenden Stelle an den (auf maximale Helligkeit eingestellten!) Monitor gehalten werden. Die Programmübertragung ist leider (je nach Programmkomplexität) keine Sache von wenigen Sekunden, sondern kann bis zu mehreren Minuten dauern. Deshalb sind Geräte empfehlenswert, bei denen der Bildschirm „liegend“ eingestellt werden kann. Lernende sind aber auch sehr kreativ im Bau von Halterungen mit Hilfe von Federmäppchen, Büchern etc., um die Bots an einen vertikalen Bildschirm zu halten, ohne sie mit den Fingern permanent fixieren zu müssen.

Ozobot evo verfügen über einen reichhaltigen Vorrat an Programmierbausteinen. Diese sind in fünf aufsteigenden Niveaustufen gruppiert. In der ersten gibt es ausschließlich Bausteine nur mit Symbolen ohne jeglichen Text. In den nächsten Stufen ist von einfachen Bewegungsbausteinen und Aktorensteuerungen sowie bedingten Verzweigungen und Wiederholungen (mit Zähler oder Bedingung) bereits eine Auswahl vorhanden, mit der komplexe Programmgestaltungen möglich sind. In den letzten Stufen gibt es dann Bausteine für selbst definierte Variablen, für mathematische und logische Operationen bis hin zu komplexen Sensorabfragen sowie für das Arbeiten mit Feldern und Unterprogrammen. Auch Bausteine zur variantenreichen Programmierung von Linienfolgern sind vorhanden.

Innerhalb eines Programms ist auch die Mischung von Blöcken verschiedener Niveaustufen möglich. Um alle algorithmischen Grundstrukturen im Grundlagenunterricht zur

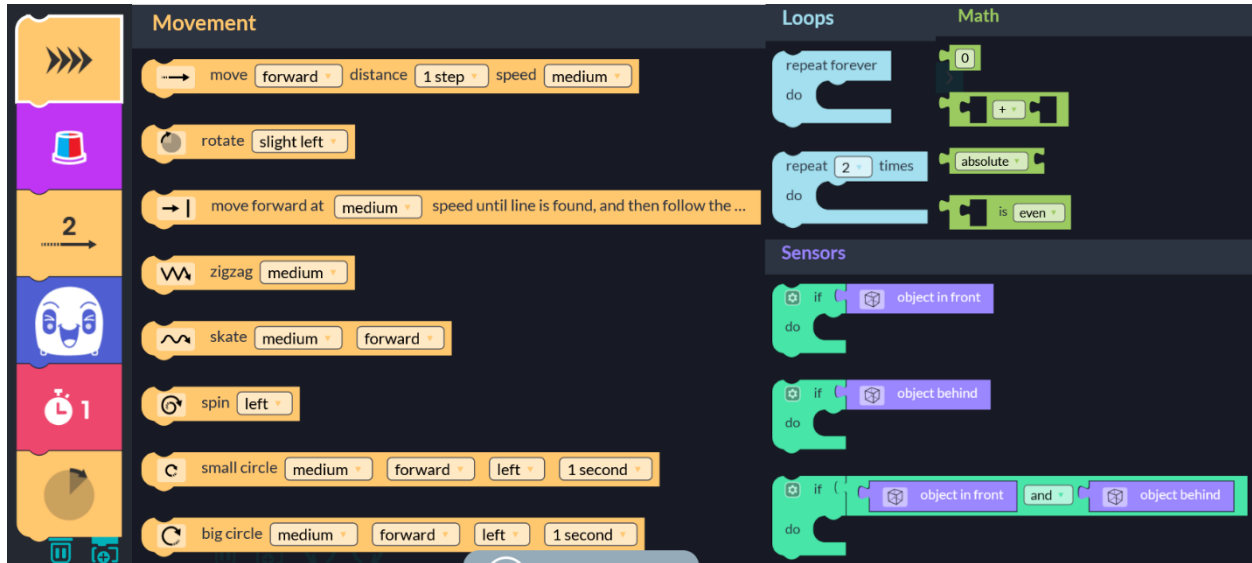


Abbildung 8: Auswahl von OzoBlockly-Befehlen verschiedener Niveaustufen

Algorithmik/Programmierung umzusetzen, sind die Niveaustufe 3 – ergänzt mit ein paar einzelnen Blöcken aus Niveaustufe 4 – bereits völlig ausreichend.

Das reichhaltige Angebot an Programmierbausteinen lässt sich gut zur Binnendifferenzierung verwenden, indem Aufgaben mit jeweils nur einer bestimmten Auswahl von Bausteinen bewältigt werden müssen. Damit niemand von den Lernenden „abgehängt“ wird, empfiehlt es sich für die Lehrkraft regelmäßig „Hilfsprogramme“ (Zwischenstände) zur Verfügung zu stellen, mit Hilfe derer die Lernenden in Problemstellungen ggf. wieder neu einsteigen können.

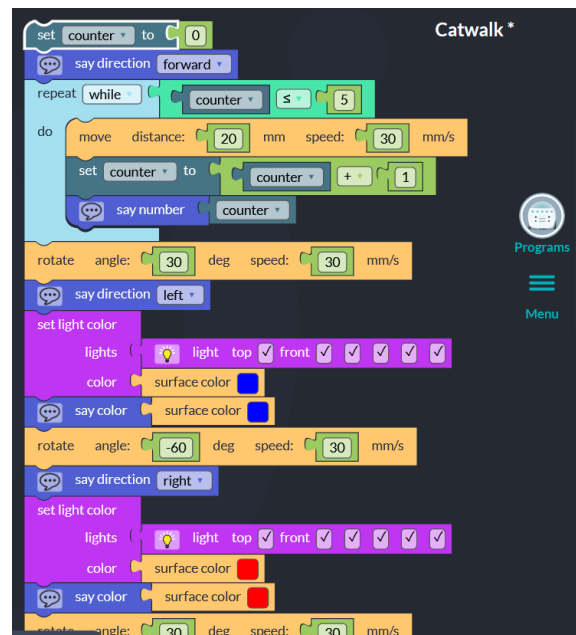


Abbildung 9: Programm zu "Model auf dem Laufsteg"

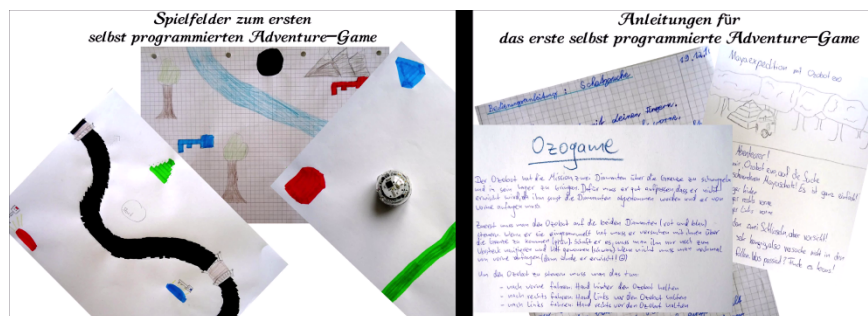


Abbildung 10: Ozobot evo - Adventure Game (Planung und Vorbereitung)

Als Einstieg haben sich einfache Programme bewährt, welche die Lernenden in ihrer Kreativität nicht einschränken. Zum Beispiel bei der Aufgabenstellung, den Ozobot zu einem „Model auf dem Laufsteg“ zu machen, lernen die Programmierenden den Umgang mit den einzelnen Befehlsblöcken und setzen bereits Wiederholungen (und die schnelleren sogar Variablen) ein. Zusätzlich benötigen sie bedingte Anweisungen und den Einsatz der Abstands-Sensoren bei Aufgaben wie dem „Besenwagen“, bei dem der Ozobot während einer Bewegung von einem vorgegebenen Start zu einem vorgegebenen Ziel alle Hindernisse (als Zubehör erhältlich) aus dem Weg räumt.

Begeistert stürzten die Lernenden sich am Ende der Unterrichtseinheit auf die komplexe Aufgabenstellung „Entwicklung eines einfachen Adventure-Games“. Der Ozobot als Hauptakteur des Spiels muss in einer selbst auf Papier gestalteten Welt verschiedenfarbige Schlüssel einsammeln, bevor er den Schatz aus seinem Versteck heben kann. Hierbei soll er unter anderem mit Hilfe seiner Abstandssensoren durch den Einsatz der menschlichen Hand berührungsfrei gesteuert werden. Nach dieser Festlegung des groben Szenarios mussten die Lernenden zunächst mit Hilfe der Ozobot-Farbstifte ein eigenes Spielfeld auf Papier gestalten und eine genaue Spielanleitung erstellen.

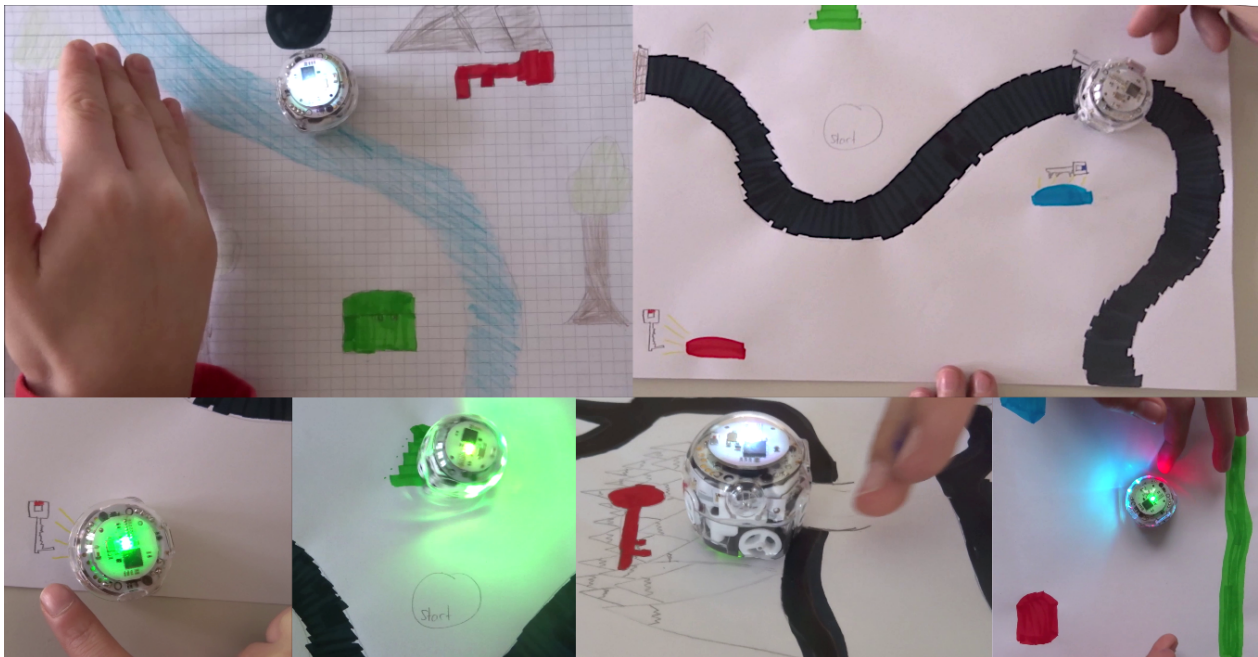


Abbildung 11: Ozobot-Adventure-Game - Spielimpressionen

Anschließend schrieben sie das passende Programm und spielten abschließend mit ihren Bots das selbst erstellte Spiel. Bei manchen gab es auch eingebaute Fallen (Überlebenspunkte wurden reduziert) oder Krokodile im Fluss (vorzeitiges Spielende). Andere perfektionierten das „Finden eines farbigen Schlüssels“ dadurch, dass der Bot nach der Erfolgsmeldung erst



Video zum „Ozobot-Adventure“

einmal das Schlüsselfeld automatisch mit einer Richtungsänderung verließ, bevor er auf weitere menschliche Steuerung wartete. Interessant war die Idee der Lernenden, die Spielfelder (und Anleitungen) untereinander auszutauschen und zu testen, ob die auch mit den nicht speziell dafür erstellten Programmen gespielt werden können.

Hinsichtlich des Aspektes des fächerübergreifenden Einsatzes der verwendeten Geräte bieten Calliope und Ozobot zwar einzelne Möglichkeiten, die aber (bei Calliope auf Grund der lediglich zwei Grove-Konnektoren) eingeschränkt sind. Hierfür eignet sich mehr der Einsatz von Geräten, wie sie im nächsten Abschnitt vorgestellt werden.

Arduino Uno

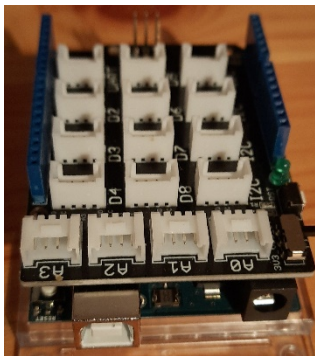


Abbildung 12: Arduino Uno mit Base-Shield

Der in diesem Abschnitt vorgestellte Mikrocontroller Arduino ist für Einsteiger geeignet und in vielen verschiedenen Board-Varianten erhältlich. Der Arduino ist ein leistungsstarker, robuster Mikrocontroller, der einfach zu programmieren ist. Er ist für Jugendliche sehr geeignet, um mit Elektronik und Programmierung zu beginnen. Alle Arduino-Boards stellen digitale Input- und Output-Pins des Mikrocontrollers zur Nutzung für elektronische Schaltungen zur Verfügung, womit er auch im Anfangsunterricht Physik für einfache elektrische Schaltungen und für die Vermittlung von Technikgrundlagen gut eingesetzt werden kann. Der Autor dieses Artikels lernte den Arduino und dessen Einsatzmöglichkeiten im

Informatikunterricht bei der Fachschaftsleitungstagung für das Fach Informatik an Gymnasien in Oberasbach (Mittelfranken) im Jahr 2018 kennen. Hierbei wurden vom Referenten Dr. Matthias Ehmann (Universität Bayreuth) Arduino Uno eingesetzt. Basierend auf den zur Verfügung gestellten Materialien (u. a. M. Ehmann (2017)) wurden die in diesem Abschnitt dargestellten Einsatzmöglichkeiten konzipiert. Deshalb wird hier nur die Board-Variante Arduino Uno betrachtet. Zusätzliche Zubehör-Sets, wie z. B. das Seeed Grove Starter Kit erweitern dessen Grundausstattung mit einem Base-Shield und 14 zusätzlichen Arduino-Sensoren sowie Zubehör. Alle Materialien lassen sich auch von Anfängern leicht beherrschen. Durch die Vielzahl von Sensoren (Helligkeit, IR, UV, Sound, Berührung, Abstand ...) und Aktoren (LEDs, LCD-Anzeige, ...) sind kreative Möglichkeiten gegeben und Binnendifferenzierung gut zu bewältigen.

Beschreibungen von vielfältigen Einsatzmöglichkeiten im Informatik-Unterricht finden sich z. B. beim Sinus-Projekt zur Steigerung der Effizienz des mathematisch-naturwissenschaftlichen Unterrichts (Sinus-Projekt (2021)), dem Landesbildungsserver Baden-Württemberg (Schülerforschungszentrum Südwürttemberg (2019)) oder bei [StartHardware.org](https://www.starthardware.org), wo Stefan Hermann von grundlegenden technischen Tutorials über Unterrichtsmaterialien bis hin zu kleinen und großen Projekten mit dem Arduino gut ausgearbeitete und einsetzbare Materialien zur Verfügung stellt. Ausführliche fachwissenschaftliche Betrachtungen zu Physical Computing im Allgemeinen und mit Arduino-Boards gestaltbaren Projekten im Speziellen

finden sich z. B. in Veröffentlichungen und Vorträgen von Prof. Dr. Mareen Grillenberger (Pädagogische Hochschule Schwyz) (u. a. Przybylla (2018, 2020) und Przybylla/Grillenberger/Schwill (2019)).

Im Folgenden wird speziell auf den möglichen Fächerübergreifung zur Physik eingegangen, da in Bayern in Jahrgangsstufe 7 für das Fach Natur und Technik jeweils nur eine Wochenstunde Informatik und eine Wochenstunde Physik in der Studententafel stehen. Durch den mit dem Arduino Uno leicht erreichbaren Fächerübergreifung können Synergien genutzt werden. Als Einstieg in die Arbeit mit dem Arduino hat sich der Einsatz eines Steckbretts mit zusätzlichen LEDs und Jumper-Kabeln (vgl. Ehmann (2017)) bewährt. Bereits mit einer einzelnen LED können sowohl das physikalische Thema „einfache Stromkreise“ behandelt als auch mit kleinen Programmen die algorithmischen Grundstrukturen im Informatik-Anfangsunterricht eingeführt werden. Hierbei steigert sich die Komplexität der Programme von einer einzelnen blinkenden LED über das wechselnde Aufleuchten zweier LEDs bis hin zu einer Ampelschaltung mit drei LEDs. Eine Binnendifferenzierung wird bei letzterer zum Beispiel durch den

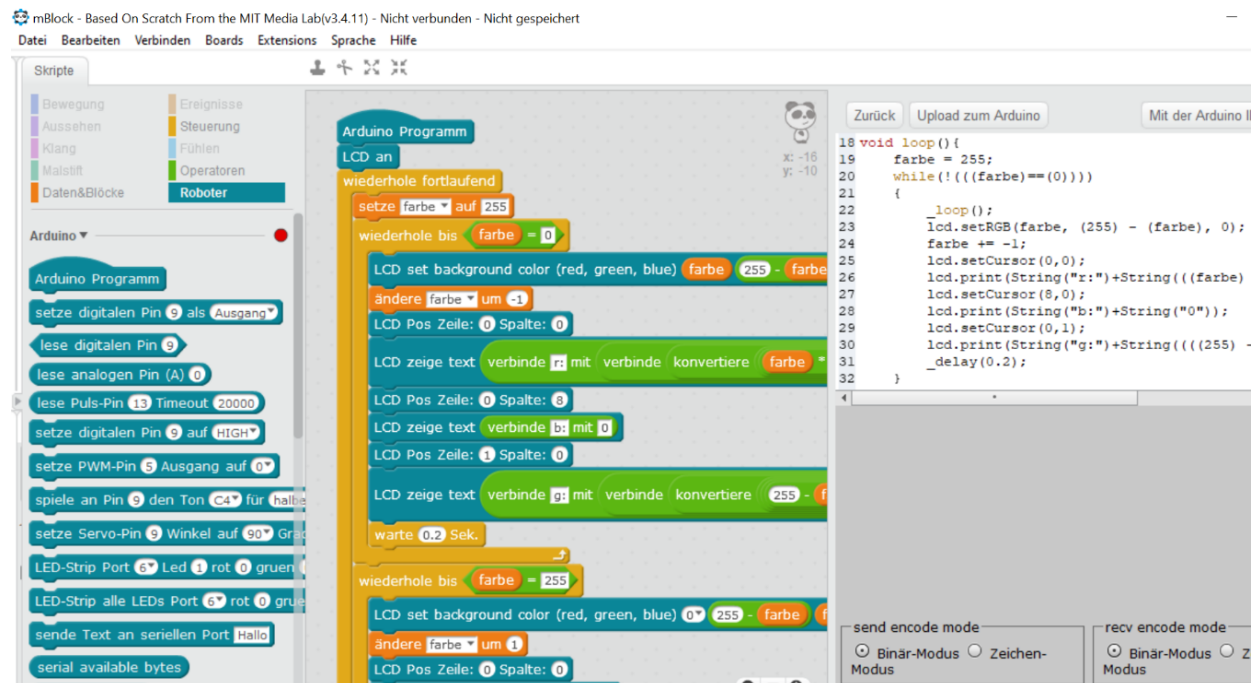


Abbildung 13: Arduino-Programm für Farbmischung im LCD-Display - Erstellung in mBlock

zusätzlichen Einsatz eines Tasters oder berührungsempfindlichen Sensors erreicht. Kreativ wurden die Lernenden bei der Entwicklung und Umsetzung eigener Ideen, wenn sie (je nach Jahreszeit) eine Weihnachtsbeleuchtung bauen und deren Steuerung programmieren sollten. Nach Vermittlung der technischen und informatischen Grundlagen kam ein Grove - LCD RGB Backlight (https://wiki.seeedstudio.com/Grove-LCD_RGB_Backlight) zum Einsatz, um weitere fächerübergreifende Themen zu behandeln. Im Lehrplan von Natur und Technik 7 – Physik stehen im Teilbereich Optik die Themen „additive Farbmischung“ und „Farbsehen“. Aus der Informatik kann beim Umgang mit dem RGB-Farbsystem das physikalische Grundlagenwissen

genutzt werden, um die Hintergrundfarbe des LCD-Displays so zu programmieren, dass kontinuierlich hintereinander alle möglichen 16,8 Millionen Farben erscheinen. Zusätzlich können hierzu auch die jeweiligen Werte der RGB-Kanäle auf dem Display angezeigt werden. Neben den algorithmischen Grundstrukturen werden hier auch selbst definierte Variablen eingesetzt. Entsprechend programmierte Arduinos lassen sich dann im Physikunterricht verwenden, um das menschliche Farbsehen von allen Lernenden selbst näher untersuchen zu lassen. Bereits ein einfacher Farbwechsler, der im Informatikunterricht für das LCD-Display programmiert wurde, ist hierfür ausreichend. Damit sind dann beeindruckende Experimente zur selektiven Reflexion (DPG-Lehrmittelkommission (2023)) möglich.

Abschluss

Alle in diesem Artikel betrachteten Werkzeuge wurden vom Autor in mehreren Jahren im Informatik-Unterricht der 7. Jahrgangsstufe (in Bayern) eingesetzt. Bei den Lernenden ließ sich keine Präferenz für eines davon feststellen, selbst wenn sie mehrere davon direkt vergleichen durften. Wichtigster positiver Aspekt war, dass sowohl bei Calliope Mini als auch bei Ozobot und Arduino mit blockbasierter Programmierung gearbeitet werden konnte. Für alle drei Werkzeuge wurde von allen Befragten genannt, dass es „toll war, dass wir auf unserem normalen Arbeitsplatz alle selbst mit den Geräten arbeiten und Aufgaben lösen konnten“. Letztendlich kann somit die Lehrkraft entscheiden, welches Werkzeug eingesetzt wird. Entscheidungskriterien können dabei sein: das Preis-Leistungs-Verhältnis (Calliope hierbei vor Arduino und Ozobot), die Vielfalt der Sensoren/Aktoren und damit verbundenen (ggf. auch fächerübergreifenden) Einsatzmöglichkeiten (Arduino hierbei vor Calliope und Ozobot) und der Faktor „Coolness“ (manchmal auch „Niedlichkeit“) (Ozobot hierbei weit vor Calliope und Arduino). Offensichtlich gilt: egal, wofür man sich entscheidet – Hauptsache es haben alle Spaß beim Lernen und Lehren.



Abbildung 14: Abschluss - Calliope, Ozobot, Arduino

Quellen

Alle Webseiten/Links wurden zuletzt geprüft am 28.08.2023.

DPG-Lehrmittelkommission (2023): Innovative Versuchsideen – Selektive Reflexion <https://www.dpg-physik.de/vereinigungen/fachuebergreifend/ag/agpp/lehrmittelkommission/versuche-1/selektive-reflexion>

Geier, G., Ebner, M. (2017): Einsatz von OZOBOTs zur informatischen Grundbildung. In: Lernen und Lehren mit Technologien: Vermittlung digitaler und informatischer Kompetenzen. Erziehung & Unterricht. Jg. 167. 7-8. S. 109-113

Geier, Gerald (2017): Feldstudie – Einheit 4 – Wegfindung
https://learninglab.tugraz.at/informatischegrundbildung/wp-content/uploads/2017/08/ozobot_einheit_4.pdf

Ehmann, M. (2017): Physical Computing mit Arduino Mikrocontrollern. In: MNUjournal, 2017-05. S. 303-310. Verlag Klaus Seeberger. 2017
https://www.mnu.de/images/publikationen/zeitschrift/Heft_5-17/MNU_5_2017_KOMPLETT.PDF

Ozobot (2023): Ozobot-Webseite. <https://ozobot.com/>

Przybylla, M. (2018): From Embedded Systems to Physical Computing: Challenges of the “Digital World” in Secondary Computer Science Education, Dissertation, Universität Potsdam
https://publishup.uni-potsdam.de/opus4-ubp/frontdoor/deliver/index/docId/41833/file/przybylla_diss.pdf

Przybylla, M. und Grillenberger, A. und Schwill, A. (2019): Networked Embedded Systems in the Physical Computing Project “Smart City”, ISSEP 2019 – 12th International conference on informatics in schools – Situation, evaluation and perspectives. University of Cyprus. ISBN 978-9925-553-27-3
<https://cyprusconferences.org/issep2019/wp-content/uploads/2019/10/LocalISSEP-v5.pdf>

Przybylla, M. (2021): Forschung und Praxis: Physical Computing im Informatikunterricht. Online-Vortrag am 20.01.2021 in einer Vortragsreihe zur Didaktik der Informatik an der Universität des Saarlandes, Deutschland
<https://informatikdidaktik.cs.uni-saarland.de/wp-content/uploads/2021/01/2021-Uni-Saarland-small.pdf>

Schülerforschungszentrum Südwürttemberg (2019): ARDUINO – Von der blinkenden LED bis zum autonomen Roboter
<https://www.schule-bw.de/faecher-und-schularten/mathematisch-naturwissenschaftliche-faecher/informatik/material/arduino/arduino-skript.pdf>

Sinus-Projekt (2021): Materialien für einen „sanften Einstieg“ in die Mikrocontroller-Programmierung
<https://www.schulentwicklung.nrw.de/materialdatenbank/material/view/5867>

StartHardware.org (2021): Projekte und Tutorials für Arduino (Stefan Hermann)
<https://starthardware.org/>

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC 4.0 zur Verfügung.

Kontakt

Dr.-Ing. Ulrich Kiesmüller

Simon-Marius-Gymnasium

Ulrich.Kiesmueller@smg-gun.de