

Lucienne-Sophie Marmé¹
Alexandra Švedkijs²
Jens-Peter Knemeyer²
Nicole Marmé²

¹didaktik-aktuell e.V.
²Pädagogische Hochschule Heidelberg

IT-Projekt „Hörtest“ mit Snap4Arduino im Physikunterricht

In den aktuellen Leitperspektiven des Bildungsplanes 2016 des Landes Baden-Württemberg wird ein „fächerintegrative Unterricht“ unter anderem auch für die informationstechnischen Grundlagen (Ministerium für Kultus, Jugend und Sport Baden-Württemberg, Medienbildung, 2016) gefordert. Mit der geplanten Bildungsreform ab dem Schuljahr 2025/2026 wird diese Entwicklung weiter verstärkt: An allen weiterführenden Schularten wird das neue Fach „Informatik und Medienbildung“ ab Klassenstufe 5 bis 11 eingeführt, wobei der Informatikanteil in den höheren Klassenstufen zunimmt (Ministerium für Kultus, Jugend und Sport Baden-Württemberg, 2024). Bundesweit zeigt sich ein verstärkter Fokus auf der Integration digitaler Kompetenzen in den MINT-Unterricht (BMBF, 2023; KMK, 2024). Mit dem Projekt „Hörtest“ wird eine konkrete Umsetzungsmöglichkeit im Physikunterricht in der Einheit Akustik gezeigt (Physik, Standards für inhaltsbezogene Kompetenzen Klasse 7/8, Optik und Akustik, 2016). Es bietet ein Beispiel für die Integration informationstechnischer Inhalte in den naturwissenschaftlichen Unterricht in Form eines Experiments. Des Hörtest ermöglicht das Experimentieren mit Frequenzen in einem für Menschen hörbaren Bereich. Alternativ kann das Projekt auch im Themenfeld E-Lehre eingesetzt werden, wobei eine Wiederholung der Akustik integriert ist. Zu den wichtigsten informationstechnischen Grundlagen gehören unter anderem Systeme zum Regeln und Steuern (z.B. Mikrocontroller) und deren Programmierung. Hierbei handelt es sich oft um komplexe Systeme (Hardware und Software), so dass die Integration in den zeitlich meist knapp bemessenen regulären Physikunterricht zunächst nicht möglich und sinnvoll erscheint. Mit dem hier vorgestellten Projekt „Hörtest“ soll eine Möglichkeit gezeigt werden, Grundlagen dieser informationstechnischen Inhalte mit einem geringen zeitlichen Aufwand in den Physikunterricht einzubauen. Hierzu wird auf ein einfaches, schnell umsetzbares Hardware-Design (Arduino-Mikrocontroller) und eine schnell zu erlernende blockbasierte Programmierung (snap4Arduino) gesetzt. Das Projekt befasst sich mit der Fragestellung, wie sich die Wahrnehmung der Frequenzen beispielsweise mit steigendem Alter ändert. Hierzu wird ein realer „Hörtest“ nachgebaut und programmiert. Über einen Lautsprecher werden unterschiedliche Frequenzen (zwischen 6 kHz und 16 kHz) in einem für den Menschen hörbaren Bereich aufsteigend als Piepton ausgegeben. Wenn eine Frequenz hörbar ist, wird der User aufgefordert, eine grüne Taste zu drücken. Falls der Ton nicht hörbar ist, dann soll eine rote Taste gedrückt werden. Die aktuell hörbare Frequenz wird dem User stetig als Information ausgegeben.

Blockbasiertes Programmieren (Snap4Arduino)

Blockbasierte Programmiersprachen erleichtern Anfänger:innen das Erlernen der fundamentalen Programmierkonzepte (Kwon, Chung, Lee, 2018). Um ein Programm zu erstellen, werden Befehle aus einer vorhandenen Bibliothek verwendet und wie ein Puzzle

verknüpft. Das zentrale Ziel von blockbasierten Programmiersprachen, wie Snap! oder Scratch ist eine Einführung in die Programmierung für Lernende ohne jegliche Vorkenntnisse (Maloney, Rusk, Resnick, Eastmond, 2010). Im Vergleich zu gewöhnlichen Programmiersprachen, die komplexe Syntax einsetzen, ermöglichen die blockbasierten Sprachen eine einfachere Interaktion mit dem Computer: Die Lernenden können sich verstärkt auf Programmierlogik konzentrieren, anstatt sich mit syntaktischen Fehlern auseinanderzusetzen (Balouktsis, Kekeris, 2016). Snap! ist eine Modifikation von Scratch und Scheme (Harvey, Mönig, 2017), die eine Reihe von Neuerungen beinhaltet und vor allem die Möglichkeit bietet, eigene Blöcke zu erstellen. Somit lassen sich mit Snap! die wichtigsten informatischen Konzepte abbilden und exemplarisch erlernen (Modrow, 2018).

Snap4Arduino ist eine Weiterentwicklung der blockbasierten Programmiersprache Snap! mit zusätzlichen Bibliotheken für die Programmierung von Mikrocontrollern und der Benutzung von Sensoren und Aktoren, die sich durch entsprechende Blöcke leicht auslesen bzw. steuern und in die Snap!-Entwicklungsumgebung integrieren lassen. Der Vorteil von snap4Arduino besteht darin, dass realitätsnahe informationstechnische Aufgaben mit einfachen, übersichtlichen Anweisungen erstellt werden können und damit die Hemmschwelle bei Schüler:innen reduziert werden kann. Snap4Arduino ermöglicht einen leichten Zugang zum Experimentieren mit Elektronik für Anfänger (Pina, Ciriza, 2017). Mit snap4Arduino besteht die Möglichkeit Programme zu erstellen, die auf den meisten weitverbreiteten Arduinoboards laufen (Pina, Ciriza, 2017). In diesem Beispiel wird der Mikrocontroller Arduino Uno verwendet. Arduino Uno ist der Klassiker der Arduino-Familie und bietet die notwendige Funktionalität, um sich mit Mikrocontrollern vertraut zu machen (Arduino Internetpräsenz). Mit snap4Arduino können Sensoren für Arduino ausgelesen werden, als Daten verarbeitet und als Ausgabe an die Aktoren übergeben werden. Das zugrunde liegende Prinzip der Eingabe - Verarbeitung - Ausgabe (EVA) kann mit diesem System im Unterricht an realen Beispielen verdeutlicht werden. Generell bietet die Kombination aus Snap! und Arduino den Vorteil verzahnende und realitätsnahe Fragestellungen im Unterricht zu behandeln, indem Signale der realen Welt über virtuelle Programme verarbeitet werden. Snap4Arduino stellt unter anderem eine Anweisung für die Ansteuerung eines Lautsprechers zur Verfügung, die in dem hier vorgestellten Projekt verwendet wird. Damit ist die Ausgabe eines Tons bestimmter Frequenz über einen vorgegebenen Zeitraum möglich.

Hardware

Der Aufbau umfasst einen Mikrocontroller Arduino UNO, ein Breadboard zum Aufbau der Schaltung, zwei Taster, einen Lautsprecher und mehrere Verbindungsleitungen (Abb. 1). Der Lautsprecher wird entsprechend der Vorgaben in der snap4Arduino-Beschreibung über einen analogen Pin, hier A0, mit dem Ground (0-Potenzial) verbunden. Durch Anlegen unterschiedlicher Potenzial-Signale an Pin A0 können so Töne unterschiedlicher Frequenzen erzeugt werden. Die beiden Taster werden mit den Digitaleingängen 8 und 9 über Pull-Down-Widerstände verbunden. Durch Betätigen der Taster wird am Pin 8 bzw. 9 ein 5V-Potenzial angelegt. Der Pull-Down-Widerstand stellt sicher, dass das Potenzial beim Loslassen der Taster wieder auf das 0-Potenzial gesetzt wird. Der hohe Widerstand $R = 10\text{ k}\Omega$ ist nötig, damit keine großen Ströme bzw. kein Kurzschluss entstehen (Brühlmann, 2015). Die Spannungsversorgung wird durch die zur Datenübertragung benötigte USB-Verbindung mit dem PC sichergestellt.

Programmierung (Software)

Prinzipiell gibt es viele Möglichkeiten ein funktionsfähiges Programm für den Hörtest zu realisieren (Abb. 2). Je nach zu Verfügung stehender Zeit, können diese mehr oder weniger komplex gestaltet werden. Auch können die Hilfestellungen der Lehrperson entsprechend unterschiedlich stark ausgeprägt sein. Nachfolgend soll eine möglichst einfache Lösungsmöglichkeit vorgestellt werden.

Nach einer kurzen Bedienungsanweisung durchläuft das Programm eine Schleife, in der die Frequenz des abgespielten Tons sukzessive erhöht wird. Dem User wird eine Informationsmeldung ausgegeben und anschließend erfolgt die Tonausgabe. Die Ausgabe beginnt mit 6000 Hz. Das Programm wartet dann auf das Betätigen eines Tasters. Die Taster liefern ein digitales Signal am digitalen Pin 8 bzw. Pin 9. Das bedeutet, dass der Tastendruck ein „high“ (true-Zustand) liefert. Falls der User keinen Ton hört, soll die rote Taste (D8) betätigt werden. Hierdurch wird beispielsweise die Anweisung „lies digitalen Pin 8“ als „true“ ausgewertet. Daraufhin wird das Skript gestoppt und es erfolgt eine Ausgabe für den User „Alles stoppt“. Falls der User stattdessen einen Ton hört und den Taster am digitalen Pin 9 drückt, wird die zweite Bedingung als „true“ ausgewertet. In diesem Fall wird die Variable *frequenz* um 2000 Hz erhöht und im folgenden Schleifendurchlauf ausgewertet.

Für sich wiederholende Anweisungen kann eine Schleifenstruktur verwendet werden. Für dieses Programm wird eine Zählschleife verwendet, die nach fünf Durchläufen endet.

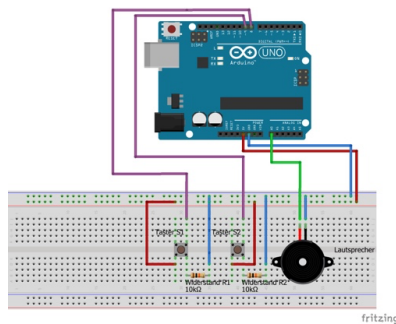


Abb. 2: Aufbau der Schaltung mit einem Lautsprecher, 2 Tastern und Arduino UNO

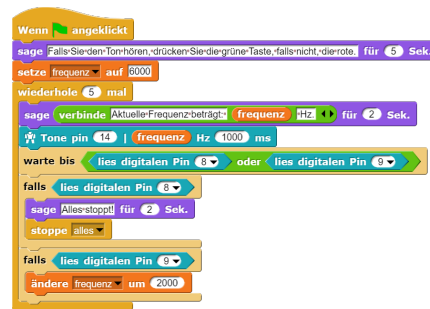


Abb. 1: Beispiel einer Umsetzung

Möglichkeiten für den Unterricht

Das beschriebene Beispiel eignet sich sowohl als ein Einführungsbeispiel in Informationstechnik als auch als eine fächerintegrierte Aufgabe. Anhand des Hörtests können die grundlegenden technischen Zusammenhänge, Verschaltung und Einstieg in die Programmierung verdeutlicht werden. Dabei ist allerdings auf didaktische Reduktion in Bezug auf Programmierung zu achten. Als fächerintegrierte Aufgabe besteht im Rahmen des Physikunterrichts beispielsweise zum Thema Akustik die Möglichkeit, dieses Projekt als Schülerprojekt zu gestalten. Idealerweise haben die Schüler:innen schon im Informatikunterricht erste Erfahrungen mit der Programmierumgebung Snap! gesammelt und/oder sich mit Mikrocontrollern beschäftigt, so dass es sich bei diesem Projekt um ein Anwendungsbeispiel handelt, in dem die Schüler:innen ihr Vorwissen anwenden und mit physikalischen Aspekten verknüpfen.

Literatur

- Arduino. (o. J.). *Arduino Uno Rev3 SMD – Produktseite*. Abgerufen am 15. Oktober 2025, von <https://store.arduino.cc/arduino-uno-rev3-smd>
- Balouktsis, I. & Kekeris, G. (2016). Learning renewable energy by Scratch programming. *Journal of Research in Education and Training, Department of Early Childhood Education, University of Ioannina*, 9, 1–10.
- Brühlmann, T. (2015). *Arduino – Praxiseinstieg*. mitp Verlags GmbH.
- Bundesministerium für Bildung und Forschung (BMBF). (2023). *Kompetenzzentren für digitales und digital gestütztes Unterrichten*. Abgerufen am 15. Oktober 2025, von https://www.bmbf.de/bmbf/de/bildung/digitalisierung-und-mint-bildung/digitaler-unterricht/kompetenzzentren-fuer-digitales-unterrachten_node.html
- Harvey, B. & Mönig, J. (2017). *Snap! Reference Manual 4.1*. Abgerufen am 15. Oktober 2025, von <https://snap.berkeley.edu/snap/help/SnapManual.pdf>
- Kultusministerkonferenz (KMK). (2024). *Empfehlung zur Stärkung der mathematisch-informatisch-naturwissenschaftlich-technischen Bildung (MINT)*. Beschluss vom 28. November 2024. Abgerufen am 15. Oktober 2025, von <https://www.kmk.org/themen/allgemeinbildende-schulen/unterrichtsfacher/mathematik-informatik-naturwissenschaften-technik-mint.html>
- Kwon, K., Chung, J. & Lee, S. J. (2018). Computational concepts reflected on Scratch programs. *International Journal of Computer Science Education in Schools*, 2(3), 1–12. <https://doi.org/10.21585/ijcses.v2i3.31>
- Maloney, J., Rusk, N., Resnick, M. & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4), 16:1–16:15. <https://doi.org/10.1145/1868358.1868363>
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg (Hrsg.). (2016). *Gymnasium, Physik – Standards für inhaltsbezogene Kompetenzen Klasse 7/8: Optik und Akustik*. Abgerufen am 15. Oktober 2025, von <http://www.bildungsplaene-bw.de/Lde/LS/BP2016BW/ALLG/GYM/PH/IK/7-8/02>
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg (Hrsg.). (2016). *Leitperspektiven und Leitfaden Demokratiebildung, Medienbildung*. Abgerufen am 15. Oktober 2025, von <http://www.bildungsplaene-bw.de/Lde/LS/BP2016BW/ALLG/LP/MB>
- Modrow, E. (2018). *Informatik mit Snap! – Snap! in Beispielen*. EMU-online. Schöden: emu-Verlag. Abgerufen am 15. Oktober 2025, von <https://emu-online.de/>
- Pina, A. & Ciriza, I. (2017). Primary level young makers programming & making electronics with Snap4Arduino. In D. Alimisis, M. Moro & E. Menegatti (Hrsg.), *Educational Robotics in the Makers Era (Edurobotics 2016)* (S. 53–64). *Advances in Intelligent Systems and Computing* (Vol. 560). Springer, Cham. https://doi.org/10.1007/978-3-319-55553-9_5
- Snap! Project. (o. J.). *Offizielle Website*. Abgerufen am 15. Oktober 2025, von <https://snap.berkeley.edu/>
- Snap4Arduino. (o. J.). *Offizielle Website*. Abgerufen am 15. Oktober 2025, von <http://snap4arduino.rocks/>