

Arbeitsmaterialien für Lehrkräfte

Kreative Ideen und Konzepte inkl. fertig ausgearbeiteter Materialien und Kopiervorlagen für einen lehrplangemäßen und innovativen Unterricht

Thema: Technik – Wirtschaft – Soziales, Ausgabe: 33

Titel: Arduino - Einführung in Mikrocontroller (47 S.)

Produktinweis zur »Kreativen Ideenbörse Sekundarstufe«

Dieser Beitrag ist Teil einer Print-Ausgabe aus der »Kreativen Ideenbörse Sekundarstufe« der Mediengruppe Oberfranken – Fachverlage GmbH & Co. KG*. Den Verweis auf die jeweilige Originalquelle finden Sie in der Fußzeile des Beitrags.

- ▶ Alle Beiträge dieser Ausgabe finden Sie [hier](#).

Seit über 15 Jahren entwickeln erfahrene Pädagoginnen und Pädagogen kreative Ideen und Konzepte inkl. sofort einsetzbarer Unterrichtsverläufe und Materialien für verschiedene Reihen der Ideenbörse.

- ▶ Informationen zu den Print-Ausgaben finden Sie [hier](#).

* Ausgaben bis zum Jahr 2015 erschienen bei OLZOG Verlag GmbH, München

Beitrag bestellen

- ▶ Klicken Sie auf die Schaltfläche **Dokument bestellen** am oberen Seitenrand.
- ▶ Alternativ finden Sie eine Volltextsuche unter www.eDidact.de/sekundarstufe.

Piktogramme

In den Beiträgen werden – je nach Fachbereich und Thema – unterschiedliche Piktogramme verwendet. Eine Übersicht der verwendeten Piktogramme finden Sie [hier](#).

Nutzungsbedingungen

Die Arbeitsmaterialien dürfen nur persönlich für Ihre eigenen Zwecke genutzt und nicht an Dritte weitergegeben bzw. Dritten zugänglich gemacht werden. Sie sind berechtigt, für Ihren eigenen Bedarf Fotokopien in Klassensatzstärke zu ziehen bzw. Ausdrücke zu erstellen. Jede gewerbliche Weitergabe oder Veröffentlichung der Arbeitsmaterialien ist unzulässig.

- ▶ Die vollständigen Nutzungsbedingungen finden Sie [hier](#).

Haben Sie noch Fragen? Gerne hilft Ihnen unser Kundenservice weiter:

[Kontaktformular](#) | ✉ Mail: service@eDidact.de

✉ Post: Mediengruppe Oberfranken – Fachverlage GmbH & Co. KG
E.-C.-Baumann-Straße 5 | 95326 Kulmbach

☎ Tel.: +49 (0)9221 / 949-204 | 📠 Fax: +49 (0)9221 / 949-377

<http://www.eDidact.de> | <https://www.bildung.mgo-fachverlage.de>

Vorüberlegungen

Kompetenzen und Unterrichtsinhalte:

- Die Schüler lernen den Mikrocontroller AT328Mega als Bauelement zur Lösung der meisten elektronischen Probleme kennen.
- Sie erhalten tieferes technisches Verständnis im Bereich „Steuern und Regeln“.
- Sie erwerben in Form eines Kleinprojektes das nötige Wissen, um ein gestelltes Problem in einer Projektarbeit lösen zu können.
- Sie stecken einfache Grundsaltungen auf einem Breadboard auf.
- Sie visualisieren Schalt- und Layoutpläne.
- Sie stellen Programmstrukturen grafisch dar.

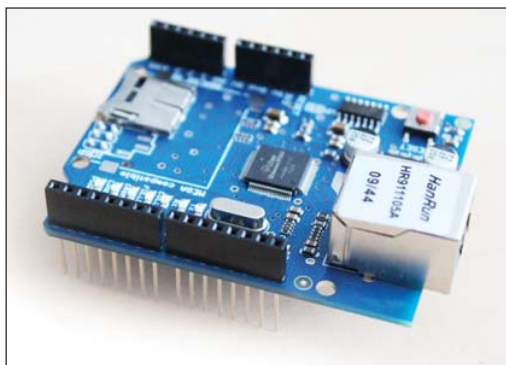
Anmerkungen zum Thema (Sachanalyse):

Wenn man heutzutage von Elektronik spricht, kommt man an **Mikrocontrollern** (auch μ Controller, μ C oder MCU) nicht mehr vorbei. Unbemerkt verrichten sie ihre Dienste in Wäschetrocknern, Herden, Unterhaltungselektronik, Steuergeräten der Kraftfahrzeuge und in nahezu allen Computer-Peripheriegeräten wie Maus und Tastatur. Überall dort, wo ein LCD-Display verbaut ist, steckt auch ein Mikrocontroller, der jenes ansteuert. Aktuell wird nahezu jedes dritte Problem aus der Elektronik mit einem Mikrocontroller gelöst.

Ein Mikrocontroller ist nichts anderes als ein **integrierter Schaltkreis**, der programmiert werden kann. Er beinhaltet aber auch all das, was einen richtigen Computer ausmacht: Recheneinheit, Speicher, I/O-Schnittschnellen, Uhr, Analog-/Digitalwandler usw. Im Gegensatz zu einem Computer befindet sich allerdings alles auf einem **kompakten Chipgehäuse**. Sobald alles programmiert wurde, benötigt der μ Controller lediglich eine Stromversorgung, um seinen Dienst zu verrichten.

Für den schulischen Einsatz haben sich inzwischen einige Systeme etabliert. Darunter findet man C-Control der Firma Conrad Elektronik, myAVR, BASIC Stamp BS1 Project Board, TinkerForge. Vor allem im Bereich „Robotik“ hat sich mittlerweile MindStorms der Firma Lego hervor getan.

Verglichen mit den genannten Systemen bleibt jedoch im Preis-Leistungs-Verhältnis die Entwicklungsplattform **Arduino** unschlagbar. Zudem ist sowohl die Hard- als auch die Softwarekomponente Open Source. Besonders die Anwendung grafischer Programmieroberflächen wie Scratch4Arduino (kurz S4A) oder ArduBlock macht das sogenannte Physical Computing auch in den unteren Klassen der Sekundarstufe interessant. Daneben gibt es zahlreiche sogenannte Shields, also Aufsteckplatinen, die den Arduino netzwerkfähig machen oder zur Ansteuerung von Motoren dienen. Außerdem gibt es dafür Hunderte fertig konfigurierter Sensoren oder Aktoren, die zum Teil auch direkt auf den Arduino aufgesteckt werden können.



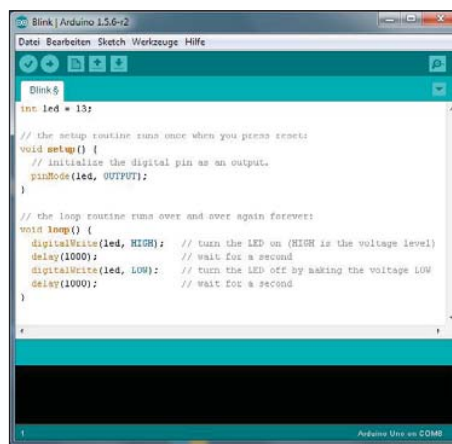
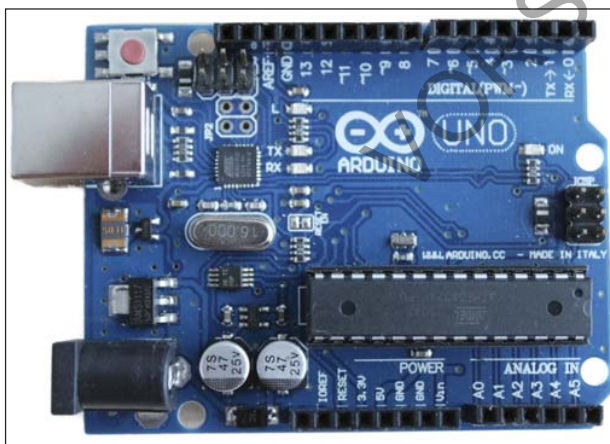
4.38

Arduino – Einführung in Mikrocontroller

Vorüberlegungen

Die **Arduino-Plattform** wurde 2005 im italienischen Ivrea entwickelt, um es Designstudenten möglichst leicht zu machen, ihre Ideen umzusetzen. Seitdem wird sie an den meisten Berufs- und Hochschulen eingesetzt, um Prototypen möglichst einfach und schnell entwickeln zu können. Auch sehr viele Bastler haben sich ihr inzwischen verschrieben und es gibt Hunderte Webseiten, auf denen jene ihre Projekte vorstellen und zum Nachbauen animieren. Auch in YouTube tummeln sich teilweise sehr gute Tutorials, in denen Usern der Einstieg in die Plattform erklärt wird.

Arduino ist also nicht nur das Board, das neben dem Mikrocontroller auch Steckleisten, Spannungsregler und USB-Anschluss beheimatet, sondern auch die dazugehörige Entwicklungsumgebung und das Framework, das per Knopfdruck den Programmcode kompiliert und per USB in den Controller hochlädt.



Dem Benutzer ist es freigestellt, den Arduino am Computer angesteckt zu lassen, wo er als Vermittler zwischen der realen und der digitalen Welt dient, oder ihn abzukoppeln und beispielsweise als Steuerung für einen Roboter-Rasenmäher zu verwenden. Natürlich kann der Arduino nicht nur über die USB-Schnittstelle mit dem PC kommunizieren. Es gibt auch Bluetooth-, WLAN- oder LAN-Schnittstellen, über die er mit Programmen wie Processing, Flash, VVVV oder Max/MSP sowie selbst geschriebener Software kommunizieren kann. So kann er beispielsweise über die Bluetooth-Schnittstelle von Smartphones Befehle empfangen, um einen Roboter zu steuern.

Warum ist Arduino für die Schule interessant?

Die wichtigsten **Vorteile**:

- Arduino ist sehr günstig. Ein Arduino UNO-Board kostet bei chinesischen eBay-Anbietern zwischen 8 und 15 Euro. Die Lieferzeit beträgt ca. vier Wochen. Deutsche Anbieter verlangen für die gleiche Ware ca. 10 Euro mehr. Dafür ist die Lieferzeit natürlich kürzer. Die Software kostet nichts und wird bei www.arduino.cc als Open Source zur Verfügung gestellt. Ratsam ist es, ein Experimentierset mit einigen Sensoren und Aktoren zu kaufen. Eine Aufstellung der wichtigsten Bauteile ist auf der **Materialliste** (siehe **M 1**) einsehbar. Der Preis hierfür liegt bei ca. 15 bis 30 Euro. Das heißt, man erhält für ca. 30 Euro ein fertig einsetzbares Schüler-Experimentierset.
- Arduino ist Open Source. Zahlreiche Entwickler erweitern die Plattform um fertige Bibliotheken, mit denen mit einfachen Methoden Servos oder Schrittmotoren angesprochen werden können. Es ist absehbar, dass auch für die Schule neben Scratch4Arduino weitere grafische Programmierumgebungen kommen werden.
- Arduino funktioniert sowohl mit Windows, Linux oder Mac OS.

Vorüberlegungen

- Arduino hat eine große Gemeinde, die stetig wächst. Es gibt fast kein technisches Problem, das noch nicht mit einem Arduino gelöst und im Internet dokumentiert wurde. Die Communities bieten Hilfe bei Problemen und verbessern Fehler in Programmcodes innerhalb weniger Stunden.
- Arduino ist einfach und noch einfacher mit Scratch. Unter anderem motiviert Arduino dadurch intrinsisch. Die Schüler merken sehr schnell, wie zeitnah sie komplexe Probleme mit wenigen Klicks lösen können. Außerdem wird der Computer so zum ersten Mal „physikalisch“. Man kann mit der Maus oder Tastatur reale Objekte, die sich neben dem Rechner befinden, steuern.

Voraussetzungen für den Unterricht

Die Arduino-Plattform ist als **Prototyping-Plattform** ausgelegt. Das heißt, alle Schaltungen werden auf einem Steckbrett aufgebaut und mit dem Arduino-Board verbunden. Sofern man keine Daten mit dem Rechner austauschen möchte, wird der PC nur zur Programmierung benötigt und das Board läuft im Anschluss mit einer 6-V-Batteriebox oder einer 9-V-Blockbatterie völlig selbstständig. Da die Schüler während der Entwicklung des Programmcodes diesen zum Testen immer wieder in den Mikrocontroller hochladen müssen, empfiehlt es sich, im Computerraum zu experimentieren. Es muss im Vorfeld die Länge des USB-Kabels beachtet werden. Falls sich keine USB-Anschlüsse auf der Vorderseite der Schülerrechner befinden, sollte es lang genug sein, um von einem Anschluss auf der Rückseite des Rechners bis zur Tischfläche zu reichen.

Die Lehrkraft sollte im Vorfeld mit Schülern die **Experimentier-Boxen** vorbereiten. Dazu gehört, dass alle Bauteile dort sorgfältig verpackt werden. Kleine Bauteile wie Widerstände oder Leuchtdioden sollten in Klarsichtbeuteln verpackt werden, größere können lose in der Box liegen. Zudem muss in jeder Box eine Stückliste vorhanden sein, sodass am Ende der Unterrichtseinheit durch die Schüler die Vollständigkeit der Box überprüft werden kann. Alternativ kann man auch ein Foto, auf dem alle Bauteile zu sehen sind, in die Box einkleben.

Außerdem muss entschieden werden, ob man mit einem Batteriehalter oder einer 9-V-Blockbatterie arbeiten möchte. Auf jeden Fall sollte man entweder am 9-V-Clip oder dem Batteriehalter zwei Anschlussmöglichkeiten parallel verlöten: Hohlstecker und zwei freie Litzen mit Steckern für das Breadboard. Man kann sich dafür einige lange **Male-Male-Steckbrücken** bei eBay bestellen, diese halbieren und mit dem Clip verlöten. Eine Alternative bietet natürlich auch ein Stecker-Netzteil, das ähnlich konfiguriert wird.

Auf jedem PC muss zudem die **Arduino IDE** (Integrated Development Environment) installiert sein. Man erhält sie in jeweils aktueller Version für Windows, Linux und Mac unter folgender URL: <http://arduino.cc/en/Main/Software>.

Es muss darauf geachtet werden, die USB-Treiber mit zu installieren!

Zudem ist es empfehlenswert, zur Visualisierung von Schaltungen die kostenlose Software „fritzing“ zu installieren. Damit können die Schüler ihre Schaltungen im Vorfeld mit dem PC entwerfen. Alle wichtigen Bauteile stehen hier bereits in Bibliotheken zur Verfügung. Auch zur Vorbereitung für die Lehrkraft, um beispielsweise Arbeitsblätter zu erstellen, ist die Software sehr zu empfehlen. Zu erhalten ist sie auf der Website des **Fritzing-Projekts**: <http://fritzing.org/download/>.

Zielsetzungen

Die Schüler sollen in Gestalt eines **Kurssystems** zunächst an einem ausgewählten Beispiel den ersten Kontakt zum Mikrocontroller knüpfen. Am Anfang werden die Schüler „nur“ nachbauen und den Programmcode abschreiben und ergänzen. Durch Verändern und Kommentieren des Codes erschlie-

4.38

Arduino – Einführung in Mikrocontroller

Vorüberlegungen

ßen sich für die Schüler neue Erkenntnisse. Auch die Gespräche mit dem Partner oder innerhalb der Gruppe dienen der Erschließung der Struktur eines Programms. Danach werden die Schüler mit kleinen **einfachen Projektthemen** konfrontiert, die sie selbst lösen sollen.

Dazu kommt jeweils die Dokumentation in Form eines Pflichtenhefts, der technischen Zeichnungen/Schaltpläne und der Kommentierung des Programmcodes bzw. der Erstellung eines Aktivitätsdiagramms.

Didaktisch-methodische Reflexionen:

Diese Unterrichtseinheit wird in fünf Schritten durchgeführt und ist durch **eigenverantwortliches** und **selbstbestimmtes Lernen** gekennzeichnet. Sie vermittelt unbewusst verschiedene Methoden aus der Wirtschaft und lädt zur Teamarbeit mit dem Partner oder in Kleingruppen ein.

Die **Projektarbeit** am Ende der Einheit ist für bereits projekterfahrene Schüler der 9. und 10. Jahrgangsstufe gedacht. Die Lehrkraft tritt selbstverständlich immer weiter in den Hintergrund und moderiert bzw. unterstützt lediglich die einzelnen Lerneinheiten.

Literatur zur Vorbereitung – Tipps für die Lehrkraft:

Internet:

- Offizielle Website des Arduino-Projekt: <http://www.arduino.cc>
- Offizielle Website des Fritzing-Projekts: <http://fritzing.org/>
- ArduinoPrX mit Video-Tutorials: <http://arduinoprj.de/>
- Beispiele der Uni Bayreuth: <http://creativecoding.uni-bayreuth.de/tools/arduino/idioms/>
- YouTube-Video-Tutorials: <http://www.youtube.com/user/MaxTechTV1>
- Website „instructables“: <http://www.instructables.com/tag/type-id/category-technology/channel-arduino/>
- Anleitungen zum Programmieren mit Arduino: http://popovic.info/html/arduino/arduinoUno_1.html

Bücher:

- Massimo Banzì: Arduino für Einsteiger, O'Reilly Verlag, Köln 2012
- Erik Bartmann: O'Reillys basics: Die elektronische Welt mit Arduino entdecken, O'Reilly Verlag, Köln 2012
- Thomas Brühlmann: Arduino Praxiseinstieg: Behandelt Arduino 1.0. mitp Professional, mitp/bhv, Heidelberg 2012
- Simon Monk: 30 Arduino Selbstbau-Projekte, Franzis Verlag, München 2012

Magazine:

- c't Hacks – Kreativ mit Technik, Heise Verlag: <http://www.heise.de/hardware-hacks/>
- Elektor: <http://www.elektor.de>

Vorüberlegungen**Die einzelnen Unterrichtsschritte im Überblick:**

1. Schritt: Kennenlernen des Arduino-Boards in Verbindung mit dem Breadboard
2. Schritt: Ansteuern von Aktoren (LED)
3. Schritt: Variablen
4. Schritt: Verzweigungen
5. Schritt: Eigene Miniprojekte

Autor: Thomas Jungkuz ist musisch/technischer Fachlehrer mit dem Schwerpunkt „Kommunikationstechnik“. Er ist hauptamtlicher Dozent am Staatsinstitut für die Ausbildung von Fachlehrern und seit 2014 Fachbereichsleiter für den Bereich „Kommunikationstechnik“. Neben Office-Anwendungen und Büroprozessen in der dortigen Studentenfirma beschäftigt er sich mit neuen Medien und Kommunikationselektronik.