

3.3. Arduino

Bei Arduino handelt es sich um Mikrocontroller-Boards, die inzwischen in vielen verschiedenen Varianten vertrieben werden. Gleichzeitig werden die zugehörige Open Source-Software und die Treiber mitgeliefert, mit Hilfe derer die Boards programmiert und ausgelesen werden können. Die Software kann direkt von der Website der Entwickler, *arduino.cc*, heruntergeladen werden.⁶ Im Gegensatz zu den ProfiLab-Programmen, die nur mit Windows funktionieren, kann Arduino auf allen gängigen Betriebssystemen (Windows, Linux, Mac) programmiert, angewendet und ausgelesen werden.

Das im Rahmen dieser Arbeit verwendete Board Arduino UNO ist beispielhaft in Abbildung 3 gezeigt. Es verfügt über 14 digitale Ein- und Ausgänge (sogenannte Pins) sowie sechs analoge Eingänge, an denen elektronische Bauteile angeschlossen werden können. Das Board wird für die Kommunikation mit dem PC mit einem Standard-USB-Kabel angeschlossen, über das auch die Stromversorgung erfolgt. Für Standalone-Anwendungen kann eine externe Spannungsquelle angeschlossen werden. Die Grundspannung des Arduino ist 5 V, an den Pins fließen Ströme bis 40 mA. Das Board arbeitet mit einer Frequenz von 16 MHz noch deutlich schneller als Digitalmultimeter und ist daher auch zur Messung schnellerer Phänomene geeignet. Dabei ist zu beachten, dass das Durchlaufen von Arduino-Programmen und Befehlen auch eine gewisse Zeit beansprucht, sodass keine Messung mit einer so hohen Ausleserate erfolgen kann. Der interne Speicher des Mikrocontrollers ist 32 kB groß, sodass nicht allzu aufwändige Messprogramme ohne Weiteres genutzt werden können.

Da sowohl die Arduino-Hardware als auch die Software *open source* sind, gibt es mittlerweile mehrere Firmen, die nachgeahmte Produkte verkaufen, die nicht in Italien oder den USA, sondern in Asien produziert werden und daher etwas günstiger sind. Ich persönlich bevorzuge die Original-Hardware; für einen Kauf im Klassensatz können jedoch andere Angebote interessant sein, beispielsweise bietet die Firma Funduino recht günstige Komplettpakete an, die ein nachgeahmtes Arduino-Board und sogar schon verschiedene Sensoren enthalten, mit denen die Schüler arbeiten können. Mit der Qualität und Lebensdauer dieser nachgeahmten Produkte habe ich bisher allerdings keine Erfahrung gemacht.

Die Vorteile bei der Verwendung von Arduino-Messsystemen im schulischen Kontext liegen auf der Hand: Sie sind sehr kompakt, günstig, vielfältig einsetzbar und dank der vielen online verfügbaren Ergänzungen, Hilfen und Bibliotheken leicht zu benutzen und an eigene Messideen anzupassen. Für physikalische Messungen ergeben sich – insbeson-

⁶<https://www.arduino.cc/en/Main/Software>, Abruf 19.07.2015

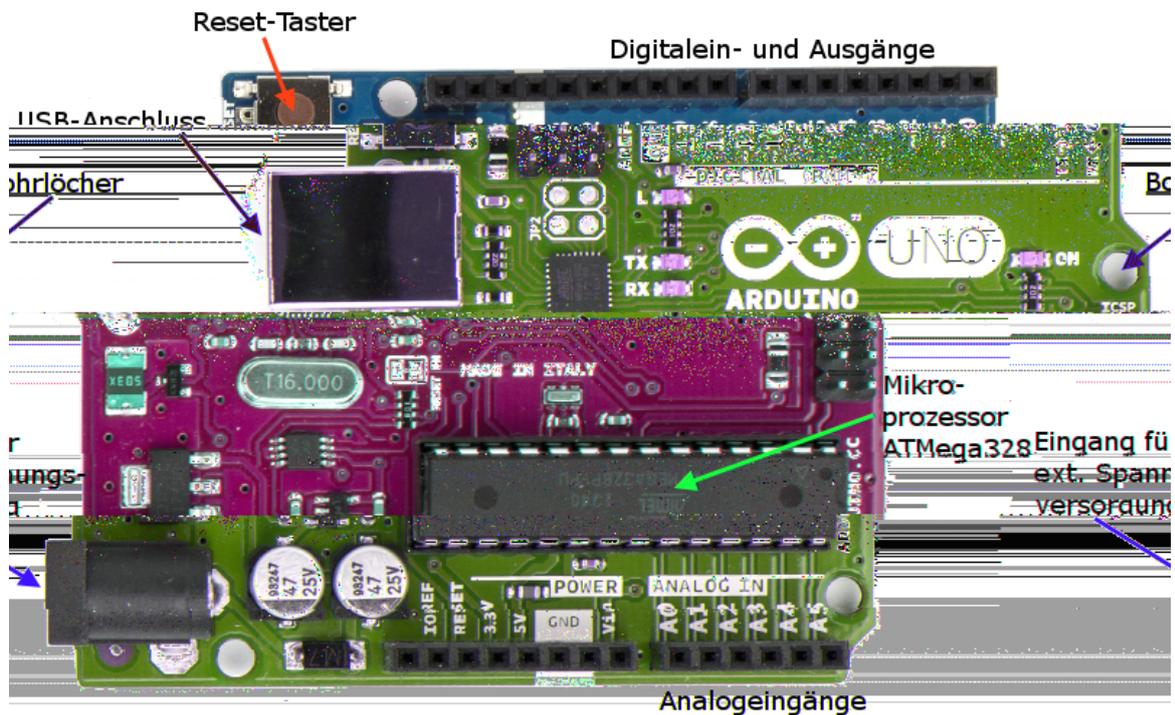


Abbildung 3: Foto (Draufsicht) eines Arduino UNO. Links befinden sich Eingänge für ein USB-Kabel und eine externe Spannungsquelle, die schwarzen Pfostenstecker oben und unten ermöglichen das Aufstecken von Erweiterungsboards (sogenannter Shields, z.B. Datalogging Shield) auf die digitalen und analogen Ein- und Ausgänge. Unten rechts ist das Herzstück des Arduino zu sehen, der Mikrocontroller ATmega328. Mit den bereits vorgefertigten Bohrlöchern kann das Board sehr einfach mit M3-Schrauben in einem Gehäuse festgeschraubt werden. Quelle: Arduino.cc (<https://www.arduino.cc/en/Main/ArduinoBoardUno>, Abruf 21.07.2015).

dere gegenüber Messungen mit dem Smartphone – einige signifikante Vorteile. In einem Gehäuse verbaut und gepolstert sind Arduino-Boards recht robust. Gerade für Würfe oder Fallbewegungen kann von Schülern nicht erwartet werden, dass sie eine Messung mit ihrem Smartphone durchführen und dabei riskieren, dass es zu Bruch geht; eine Arduino-Sensorbox allerdings kann ohne Weiteres geworfen werden, ohne dass die Schüler Angst um ihr teures Eigentum haben müssen. Mit entsprechenden Sensoren können darüber hinaus deutlich größere Beschleunigungen gemessen werden als mit Smartphones – mit modernen Beschleunigungssensoren sind sogar Beschleunigungen bis zum 16-fachen der Erdbeschleunigung messbar. Dies kann für einige Messungen sehr hilfreich sein, beispielsweise bei Achterbahnfahrten oder schnellen Rotationsbewegungen. Auch bei Magnetfeldmessungen können mit den passenden Sensoren deutlich größere Messbereiche abgedeckt werden als mit dem Smartphone. Da die Messprogramme selbst entwickelt und nicht auf einem Smartphone installiert werden müssen, fallen keine Kontrollen und eventuell sogar Blockaden der Software durch den Apple App Store oder den Google Play Store an. Die Verteilung von Messprogrammen an die Schüler ist ohne Downloads von Drittanbietern möglich und verzichtet darüber hinaus vollständig auf Werbung.

Selbstverständlich ist der Bau einer Sensorbox vor der Verwendung mit einiger Arbeit verbunden. Die Investition zahlt sich aber meiner Meinung nach insofern aus, dass danach die Freiheiten in der Benutzung sehr viel größer sind als mit vorgefertigten Messlösungen oder Smartphones. Gerade, wenn eine Sensorbox mit austauschbaren Sensoren und Deckeln für verschiedene Messungen angefertigt wird, ist der Nutzen für den Unterricht sehr groß, da für die Vorbereitung des Experiments lediglich ein Sensorboard aufgesteckt bzw. der Deckel gewechselt und ein anderes Arduino-Programm hochgeladen werden müssen. Da der Grundaufbau immer gleich bleibt und die Bedienung von ProfiLab-Ausleseprogrammen recht intuitiv ist, muss man sich nicht ständig in andere Benutzerhandbücher einlesen.

Kommunikation mit dem PC und Datenauslese

Die im Rahmen dieser Arbeit entwickelten Arduino-Programme kommunizieren über eine sogenannte Serielle Schnittstelle mit dem PC. Dafür müssen die mit der Arduino-Software gelieferten Treiber installiert werden. Verfügt die Schule nicht über Windows-PCs, sodass die ProfiLab-Programme nicht genutzt werden können, oder sind alternative Auswertungsprogramme erwünscht, können die von den Arduino-Boards gesendeten Daten auch anderweitig ausgelesen werden. Die Form, in der die Daten an den PC übertragen werden, ist jeweils in der Beschreibung der Programme angegeben, sodass

der Endverbraucher frei darüber entscheiden kann, wie er die Messprogramme verwendet. Die Befehle, die in den jeweiligen Programmen zur Steuerung des Arduino-Boards genutzt werden, sind in einer auf der Projektwebseite, <http://schulmaterial.physi.uni-heidelberg.de>, hinterlegten Kurzanleitung erklärt.

Inzwischen gibt es auch kleine Mikrocontroller-Boards, die das Senden und Empfangen von WLAN-Signalen ermöglichen, beispielsweise der Mikrocontroller ESP8266 vom Hersteller Espressif. Dadurch wären auch drahtlose Live-Messungen denkbar, jedoch nicht für sehr schnelle Vorgänge, da die Übertragung eine gewisse Zeit beansprucht. Hierfür sind zudem tiefergehende IT-Kenntnisse Voraussetzung.

Mobile Messungen mit Datalogging Shield

Um auch mobile Messungen durchführen zu können, wird eine Möglichkeit benötigt, die Messdaten möglichst zusammen mit einer Echtzeit-Zeitmarke des Aufnahmezeitpunktes zwischenzuspeichern, um sie später am PC auswerten zu können. Hierzu bietet sich ein sogenanntes Datalogging Shield von Adafruit Industries an, das gleichzeitig eine Schnittstelle für SD-Karten und eine Echtzeituhr bereitstellt und sich daher sehr gut für mobile Messungen eignet. Auf diesem Shield kann auch direkt eine Schaltung mit Sensoren verlötet werden, da ein Bereich mit Lochraster zur Verfügung steht (siehe Abbildung 4); um eine flexiblere Handhabung mit verschiedenen Sensoren zu ermöglichen, wurden diese jedoch auf zusätzlichen kleinen Lochrasterplatinen verlötet, die dann auf das Datalogging Shield aufgesteckt werden können. In Abbildung 4 sind ein auf ein Arduino-Board aufgestecktes Datalogging Shield und seine Komponenten gezeigt.

Das Board muss für Messungen unterwegs zusätzlich mit einer mobilen Spannungsquelle versorgt werden. Diese externe Gleichspannungsquelle muss eine empfohlene Spannung von 7-12 V liefern, die auf dem Board durch einen Spannungsregler auf 5 V reduziert wird. Aus diesem Grund bieten sich 9 V-Blockbatterien an, da sie relativ günstig sind und relativ lange halten; hierbei muss aber beachtet werden, dass die Spannung nicht unter etwa 7.5 V fällt, da andernfalls die Programme nicht mehr stabil laufen – vor einer Messung sollte die Batterie also ausgetauscht oder zumindest eine Ersatzbatterie bereitgehalten werden. Damit bei mobilen Messungen kein Stecker von der Außenseite an das Board angeschlossen werden muss, wurde die Batteriehalterung hier direkt von unten an das Arduino-Board angelötet und ein Schiebeschalter zum Abklemmen der Spannungsquelle zwischengeschaltet. So sind sowohl kabelgebundene Messungen als auch mobile Messungen mit der selben Sensorbox möglich, was erneut für eine Reduktion der Kosten für die Sensorboxen sorgt. In Abbildung 5 ist eine Sensorbox zur mobilen

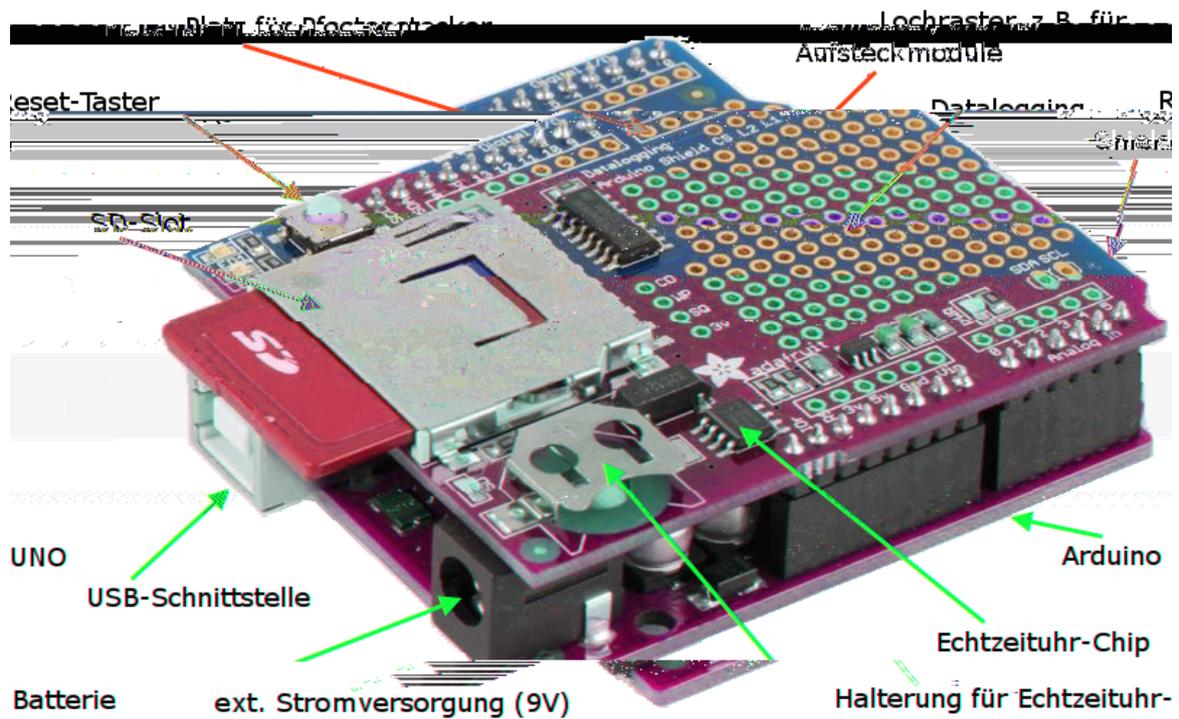


Abbildung 4: Foto eines auf einem Arduino UNO aufgesteckten Adafruit Datalogging Shields. Werden auf das Shield erneut Pfostenstecker aufgelötet, können darauf Sensor-Aufsteckboards gesteckt werden. Dadurch können mit dem selben Arduino mit Datalogging Shield verschiedene Sensoren verwendet werden. Die Sensoren können alternativ auch direkt auf dem Lochraster verlötet werden. Durch die Echtzeituhr und die Schnittstelle für SD-Karten ist ein solches Modul für mobile und kabellose Messungen sehr gut geeignet. Quelle: Adafruit Industries (<https://learn.adafruit.com/adafruit-data-logger-shield>, Abruf 21.07.2015).

Messung von Beschleunigungen gezeigt. Der Deckel mit den Löchern, durch die die Taster betätigt und die Statusleuchten gesehen werden können, wurde hier entfernt. Auch die Schaumstoffpolsterung, mit der die aufgesteckten Komponenten und die Batterie vor einem Herausfallen bei starken ruckartigen Bewegungen gesichert werden sollten, ist hier nicht gezeigt, da sie die anderen Komponenten verdecken würde.

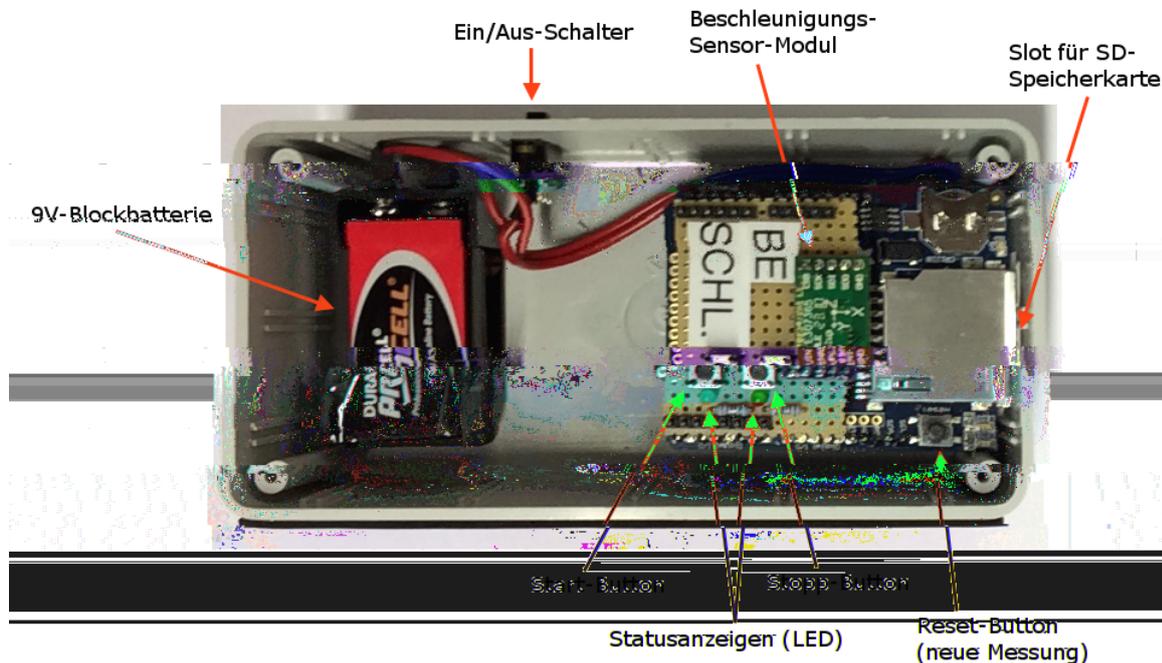


Abbildung 5: Foto der Sensorbox mit Beschleunigungssensor-Modul. Um nicht angeschraubte Teile wie die Batterie oder das aufgesteckte Sensor-Modul gegen Ablösen beim Aufprall zu sichern, sollten vor Anschrauben des Deckels Schaumstoffstücke in die Sensorbox gelegt und letztere zusätzlich mit Luftpolsterfolie umwickelt werden. Das Drücken der Start- und Stopp-Knöpfe kann auch noch durch die Polsterung hindurch erfolgen, indem ein dünner Stift durch ein kleines Loch in der Polsterung auf den jeweiligen Knopf gedrückt wird.

Warum Arduino und nicht z.B. Raspberry Pi?

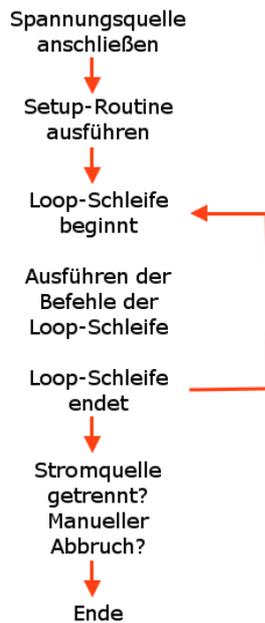
Das Grundprinzip der Arduino-Programme ist, dass nach jedem Start und einmaligem Durchlaufen einer Setup-Funktion eine Schleife so lange immer wieder durchlaufen wird, bis die Stromversorgung unterbrochen oder das Board durch die Software zum Anhalten gezwungen wird. Dieses Prinzip ist in Abbildung 6 schematisch und mit einem Programm

ohne auszuführende Inhalte verdeutlicht. Raspberry Pi hingegen arbeitet mit einem eigenen Betriebssystem, das von einer Speicherkarte geladen wird. Die Programmierung von Messprogrammen ist zwar ähnlich einfach wie mit Arduino, das Board benötigt jedoch einen Bildschirm und Eingabemedien (z.B. Maus und Tastatur), um benutzt werden zu können.⁷ Zudem sind auf dem Board keine analogen Eingänge verfügbar. Für sehr einfache Anwendungen mit möglichst wenig Benutzerabhängigkeit kann es daher nicht genutzt werden. Die Einfachheit des gezeigten Programmablaufes und der Arduino-Befehle sowie die günstigen Konditionen haben mich dazu bewegt, das Arduino-System gegenüber anderen Systemen zu bevorzugen. Darüber hinaus gibt es eine sehr große Arduino-Community, in der verschiedene Nutzer ihre Ideen und Programme anderen Nutzern zur Verfügung stellen. Auch bieten viele Sensor-Hersteller Beispielquelltexte und Bibliotheken an, mit denen ihre Sensoren einfach in Arduino genutzt werden können.

Weitere Informationen

Die hier gezeigten Informationen zeigen nur einen sehr kleinen Ausschnitt der verfügbaren Arduino-Produkte. Je nach Anwendung und preislicher Flexibilität können auch andere Boards in Frage kommen, für die die bereitgestellten Messprogramme aber gegebenenfalls nicht ohne Weiteres funktionieren. Dennoch kann es für bastelbegeisterte Lehrer und Schüler interessant sein, sich über die weiteren Möglichkeiten zu informieren. Hierzu ist die Arduino-Website – <https://www.arduino.cc> – die beste Adresse. Dort finden sich auch weitere Informationen zu den Arduino-Befehlen, Tutorials zum Erlernen der Programmiersprache, Projektideen und viele weitere Angebote.

⁷Die Steuerung ist auch per Fernsteuerung möglich, aber komplizierter als mit Maus, Tastatur und Display.



```

1 // Anschließen einer Spannungsquelle startet das Arduino-Programm.
2
3 void setup() {
4   // Diese Funktion wird immer nach einem Neustart des Arduino-Programmes ausgeführt.
5   // Hier kann die Serielle Schnittstelle gestartet und das Programm in einen
6   // sinnvollen Ausgangszustand gebracht werden.
7   // Auch der Kontakt zu eventuellen Sensoren und Speichermedien wird hier hergestellt.
8   // Wenn die Setup-Routine beendet ist, startet die Loop-Funktion.
9 }
10
11 void loop() {
12   // Der Quelltext, der sich in dieser Funktion befindet, wird so lange wiederholt,
13   // bis die Stromversorgung getrennt oder das Programm durch einen entsprechenden
14   // Befehl angehalten wird.
15   // Hier werden Messungen, das Speichern von Daten oder auch die Abfrage von
16   // Taster-Stellungen vorgenommen.
17 }
  
```

Abbildung 6: Grundprinzip eines Arduino-Programmes. Sobald das Board an eine Spannungsquelle angeschlossen wird, durchläuft es zunächst einmal eine Setup-Routine, bevor es in eine Endlos-Schleife übergeht, die nur durch Trennen der Stromversorgung oder ein Anhalten durch die Software unterbrochen wird. In dieser Schleife werden alle Messbefehle, Datenspeicherungen, Taster-Abfragen und Leuchtanzeigen-Steuerungen vorgenommen.