

Arduino-Baukasten

Zielstellung

Die Verbindung eines Arduino-Mikrocontrollerboards mit S4A eignet sich für den Entwurf interaktiver Installationen im Informatikunterricht jeder Altersstufe. Bei geeigneter Reduktion der Bauteile ermöglicht diese Hard- und Softwarekombination den Schülern sofortiges „tinkering“:

„Tinkering is what happens when you try something you don't quite know how to do, guided by whim, imagination, and curiosity. When you tinker, there are no instructions - but there are also no failures, no right or wrong ways of doing things. It's about figuring out how things work and reworking them. Contraptions, machines, wildly mismatched objects working in harmony - this is the stuff of tinkering.“
[Ban11]

Dieses experimentelle Vorgehen ist im Sinne eines motivierenden und kreativitätsfördernden Informatikunterrichts unbedingt wünschenswert. Auch die Ideen der konstruktionistischen Lerntheorie können durch einen entsprechend gestalteten Einsatz der Technik im Unterricht umgesetzt werden. Um den Schülern ein sofortiges Ausprobieren und Experimentieren zu ermöglichen, ist es notwendig, die Komponenten so zu gestalten, dass die Schüler nicht über elektrotechnische Probleme nachdenken müssen. Komponenten, die für das Verständnis des Sensors oder Aktors nicht relevant sind, werden daher in Blackboxes versteckt, welche jedoch bei Bedarf „geöffnet“ werden können, indem die Bauteile reproduziert oder die Datenblätter studiert werden. Die einzelnen Bauteile sollen außerdem so gestaltet sein, dass sie flexibel einsetzbar sind, also zum Beispiel über Verlängerungen auch in einer größeren Distanz zum Board angebracht werden können. Außerdem sollen einfache Steckverbindungen geschaffen werden, so dass die Schüler nicht mit winzigen Litzen hantieren müssen, die leicht aus den Pin-Eingängen am Arduino rutschen oder brechen. Der Baukasten soll zudem einfach um zusätzliche Komponenten und Ersatzteile erweiterbar sein. Im Folgenden werden die Komponenten des Baukastens, ihre technische Eigenschaften und die mögliche Verwendung beschrieben.

Bestandteile des Baukastens

Sensor- und Aktorenboard (Shield)


Um die Handhabung der Sensoren und Aktoren zu vereinfachen und um die Ein- und Ausgabeseite der interaktiven Installationen auch optisch klar voneinander zu trennen empfiehlt es sich, die Eingänge für die Sensoren und die Ausgänge für die Aktoren entsprechend der Verwendung mit S4A zu gruppieren. Gleichzeitig ist aber auch die Verwendung anderer Software (wie z.B. ModKit) möglich. Arduino ist nicht dafür ausgelegt, vorgefertigte Sensoren und Aktoren schnell und einfach anzustecken, was das Experimentieren unnötig verkompliziert. Daher werden auf den für die Nutzung

im Unterricht hergestellten Shields Strom, Masse und Daten immer so nebeneinander gelegt, dass die Sensoren und Aktoren mittels Steckverbindungen leicht angebracht werden können. So wird auch verhindert, dass Komponenten falsch herum angesteckt werden können.

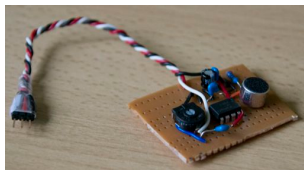
Sensoren

Die Sensoren werden so vorbereitet, dass sie sofort angesteckt und ausgelesen werden können. Die Schüler sollen kein elektrotechnisches Vorwissen benötigen, um mit dem Set arbeiten zu können. Dies schließt nicht aus, dass im Sinne eines fächerübergreifenden Unterrichts beispielsweise in den Fächern Physik oder Arbeitslehre neue Sensoren für die Baukästen konzipiert und hergestellt werden können. Von den Sensoren soll all das, was an zusätzlichen Bauteilen verwendet wird, als Blackbox vorliegen, so dass für die Schüler lediglich der Sensorkopf und der Stecker für das Sensorboard sichtbar sind. So werden beispielsweise Vorwiderstände, welche u.A. notwendig sind, um über die Datenleitung den Zustand eines Tasters abzufragen, für die Schüler nicht sichtbar sein, sondern hinter einer Isolation verschwinden. In Tabelle 1 werden die Sensoren, welche Bestandteile des Baukastens sind, näher beschrieben.

Tabelle 1: Sensoren und ihre Verwendung

| <i>Sensor</i> | <i>Details</i> |
|--|--|
| Helligkeitssensor  | Ein Helligkeitssensor ist ein lichtabhängiger Widerstand, welcher in Verbindung mit einem Pull-Down-Widerstand über Arduino ausgelesen werden kann. Der Widerstand des Sensors ändert sich in Abhängigkeit von der Helligkeit des einfallenden Lichts. Je dunkler die Umgebung ist, desto höher ist der Widerstand des Sensors. Experimentell können ein oder mehrere Schwellwerte bestimmt werden, die einer bestimmten Helligkeit entsprechen. Diese werden dann genutzt um entsprechende Aktionen auszuführen. |
| Temperatursensor  | Temperatursensoren sind Helligkeitssensoren vom Aufbau sehr ähnlich. Ein Pull-Down-Widerstand ist bei den in diesem Set verwendeten Sensoren jedoch nicht nötig, da sie bereits über einen Datenpin verfügen. Der Widerstand bzw. die Durchlassspannung der Temperatursensoren verändert sich proportional zur Umgebungstemperatur. Werden 0°C gemessen, entspricht dies dem maximalen Widerstand des Sensors. Somit wird am Eingangspin der Wert 0 gelesen. Werden 100°C gemessen, ist der Widerstand des Sensors sehr gering und am Eingangspin liegt die maximale Spannung an, was dem Wert 1023 ent- |

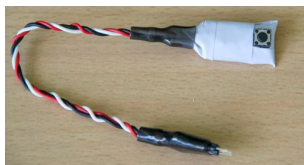
Soundsensor



spricht. Dementsprechend können auch hier Grenzwerte errechnet werden, die einer bestimmten Umgebungstemperatur entsprechen und mit denen im weiteren Verlauf gearbeitet werden kann. In der softwareseitigen Umsetzung empfiehlt es sich, Durchschnittswerte als Basis für die Temperaturberechnung zu verwenden, da Schwankungen auftreten können, die somit ausgeglichen werden.

Ein Soundsensor lässt sich realisieren, indem eine einfache Mikrofonkapsel mit einem Vorverstärker betrieben wird. Der Aufbau einer Verstärkerschaltung ist relativ aufwendig und resultiert im Eigenbau in einer recht unhandlichen Form, da die verwendeten Kondensatoren recht viel Platz einnehmen. Es wäre daher eine Überlegung wert, vorgefertigte Soundsensoren zu verwenden, auch wenn diese preislich nicht mit selbst hergestellten Sensoren konkurrieren können. Softwareseitig können diese Sensoren im Prinzip ähnlich den Temperatur- und Helligkeitssensoren verwendet werden. Man findet zunächst experimentell den Sensorwert für die aktuelle Geräuschkulisse heraus, um dann Abweichungen feststellen zu können. Ebenso weisen bestimmte Geräusche auch bestimmte Sensorwerte auf. Dies kann ebenfalls experimentell erfasst, anschließend analysiert und schließlich genutzt werden.

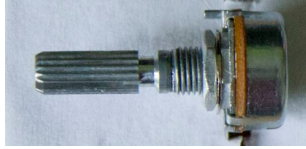
Schalter und Taster



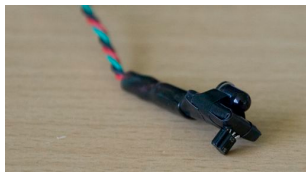
Bei den beigelegten Schaltern handelt es sich eigentlich um Taster, welche mittels Pull-Down-Widerstand mit einer Datenleitung versehen wurden. Somit ist ihr Zustand auslesbar, was wiederum zur Folge hat, dass sie über die Software gesteuert auch die Funktion eines Schalters einnehmen können. Hierzu muss der entsprechende Eingangspin beobachtet werden. Bei einer Zustandsänderung (also wenn der Taster gedrückt wird), ändert sich der Wert einer Schaltervariablen. Hierbei muss eine geeignete Verzögerung eingeplant werden, damit kein ständiges An- und Ausschalten erfolgt, sofern dies nicht erwünscht ist. Alternativ können natürlich auch Schalter verwendet werden, die in der geschalteten Position verharren. Die Tasterfunktion des Sensors bleibt selbstverständlich erhalten.

Potentiometer

Ein Potentiometer ist ein veränderbarer Widerstand. Potentiometer eignen sich sehr gut, um Aktoren manuell und scheinbar stufenlos zu steuern (z.B. Helligkeit, Lautstärke,



Näherungssensor (IR)



Geschwindigkeit etc.). Je nach Einstellung (Höhe des Widerstandes) wird über S4A ein Wert zwischen 0 und 1023 eingelesen. Dieser kann dann verarbeitet werden. Für diese Zwecke wurde ein Drehpotentiometer an der Eingangsplatine angebracht.

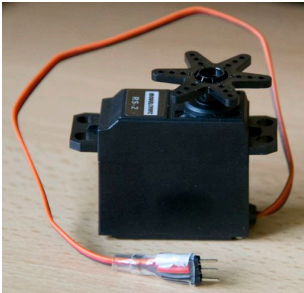

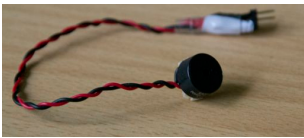
Der hier verwendete Näherungssensor besteht aus einer Infrarot-Diode und einem Infrarot-Empfänger, welche durch einen Sichtschutz voneinander getrennt sind. Die IR-Diode sendet Licht, welches – sofern ein Objekt in Reichweite ist – reflektiert und vom IR-Empfänger wahrgenommen wird. Der Sensor erkennt weiße Gegenstände ab einem Abstand von ca. 5-6 cm. Schwarze Gegenstände nimmt er ab ca. 1 cm Abstand wahr. Wird ein Objekt erkannt, liegt das Signal 0 am entsprechenden Eingangspin, da der Empfänger nach der Methode „low on detection“ arbeitet. Der Näherungssensor kann beispielsweise als sehr einfacher Bewegungsmelder eingesetzt werden oder beim berührungsfreien schalten hilfreich sein.

Eine Baukastenerweiterung könnte zahlreiche weitere Sensoren enthalten, wie beispielsweise Kippschalter, Hall-, Druck-, und Berührungssensoren, Lichtschranken, Vibrations- und Ultraschallsensoren.

Aktoren

Auch beim Entwurf der Aktoren wird so verfahren, wie schon für die Sensoren beschrieben. In Tabelle 2 werden jene Aktoren näher beschrieben, welche Bestandteil des Baukastens sind.

Tabelle 2: Aktoren und ihre Verwendung

| <i>Aktor</i> | <i>Details</i> |
|---|---|
| <p>Standard-Servo</p>  | <p>Als Standard-Servos werden solche Servomotoren bezeichnet, die sich – im Gegensatz zu CR-Servos – nur in einem bestimmten Winkel bewegen können. Die in diesem Set verwendeten Servos lassen sich von 0° bis 180° bewegen und werden in S4A über die Motorblöcke 8 und 12 angesteuert. Je nach Anwendung kann es sinnvoll sein, den Anfangspunkt (0°) oder den Mittelpunkt (90°) als Basis zu nutzen. Der Betrieb über eine zusätzliche Stromquelle ist notwendig, da die Stromversorgung über USB für die Servos nicht ausreicht.</p> |
| <p>CR-Servo</p> | <p>Bei CR-Servos handelt es sich um modifizierte Standard-Servos, die sich kontinuierlich drehen können. Sie werden in S4A über die Motorblöcke 4 und 7 gesteuert. Man kann sie an- und abschalten sowie die Drehrichtung (im/gegen den Uhrzeigersinn) angeben. Der Betrieb über eine zusätzliche Stromquelle ist notwendig, da die Servos ansonsten nicht ausreichend Strom bekommen würden.</p> |
| <p>LEDs</p>  | <p>Die hier verwendeten LEDs wurden allesamt mit Vorwiderständen ausgestattet, um frühzeitigen Defekten entgegenzuwirken. Diese LEDs können digital (on/off) oder über Pulsweitenmodulation (PWM) angesteuert werden. PWM simuliert eine analoge Ansteuerung und bewirkt, dass sich die Helligkeit der LED scheinbar stufenlos steuern lässt.</p> |
| <p>Piezo-Summer</p>  | <p>Piezosummer sind einfachste Lautsprecher, mit denen Klick- und Piepstöne erzeugt werden können. Über PWM lässt sich die Lautstärke des erzeugten Tons steuern. Das Erzeugen verschiedener Tonhöhen, wie es zum Abspielen von Melodien benötigt werden würde, ist mit S4A leider sehr kompliziert.</p> |

Eine Baukastenerweiterung könnte als zusätzliche Aktoren beispielsweise Vibrationsmotoren und verschiedenartige Displays enthalten.

Zubehör

Neben den Shields, Sensoren und Aktoren enthält der Baukasten noch das folgende Zubehör, welches zum Experimentieren benötigt wird. (siehe Tabelle 3)

Tabelle 3: Zubehör im Arduino-Baukasten

| <i>Bestandteil</i> | <i>Details</i> |
|---|---|
| Arduino Uno | Der Baukasten enthält auch eine aktuelle Arduino-Version. Dies ist zur Zeit Arduino Uno. |
| Verlängerungskabel  | Es werden einige Verlängerungskabel bereitgestellt, um Distanzen zwischen Arduino und Sensoren bzw. Aktoren zu überbrücken. Hierfür werden je Kabel drei Drähte an die entsprechenden Stecker angebracht, welche die gleichen farbigen Markierungen erhalten, wie die anderen Bauteile. |
| Batteriehalter  | Es wird ein Batteriehalter mit 2,1mm-Hohlstecker bereitgestellt, um die interaktiven Objekte und Installationen mit einer 9V-Block-Batterie betreiben zu können, wenn nötig. |
| Netzteil | Es wird ein 9V-Netzteil beigelegt, um z.B. bei der Benutzung von Servos nicht auf Batteriebetrieb angewiesen zu sein. |
| USB-Kabel | Für die Kommunikation mit dem Computer wird ein USB Kabel (Typ B) beigelegt. |
| Anleitung | Eine Bedienungsanleitung für den Baukasten inklusive der detaillierten Beschreibung von Beispielprojekten soll ebenfalls Bestandteil des Baukastens sein. |
| Datenblätter | Die Datenblätter der einzelnen Bestandteile des Baukastens werden für den Nachbau bereitgestellt. Somit wird es den Nutzern ermöglicht, selbst Ersatzteile herzustellen, ohne diese vorab konzipieren zu müssen. |
| Software | Es wird darauf verzichtet, dem Baukasten einen Datenträger mit der notwendigen Software beizulegen, da diese kostenlos im Internet verfügbar ist und es zudem empfohlen wird, stets mit der aktuellen Version zu arbeiten. (siehe [Ard12], [Con11]) |

Literaturverzeichnis

- [Ard12] Arduino (2012): *Arduino - Software*. <http://arduino.cc/en/Main/Software>, zuletzt geprüft am 13.April.2012.
- [Ban11] Banzi M. (2011): *Getting Started with Arduino*. 2nd Edition. Aufl. Sebastopol, CA: O'Reilly Media / Make.
- [Con11] S4A (2012): *Scratch for Arduino*. <http://seaside.citilab.eu/scratch/arduino>, zuletzt geprüft am 10.April.2012.