

Jens-Uwe Grabowski

Einsatz des Arduino Uno im Unterricht im Fach Informatik

Bericht zum Seminar
Didaktik ausgewählter Aspekte

Technische Universität Dresden

SS 2019

Inhaltsverzeichnis

Herleitung.....	3
Einordnung.....	4
Andere Werkzeuge.....	4
Vorhandene Lehr-Lern-Materialien.....	4
Motivation.....	6
Reihenplanung.....	7
Stunde1: Grundlagen eines Aufbaus mit dem Arduino.....	8
Stunde2: Arbeit mit verschiedenen Aktoren und Sensoren.....	9
Stunde3: Arbeit mit verschiedenen Programmelementen.....	10
Lehr-Lern-Material.....	12
Demonstration eines Aufbaus aus Schaltung und Programm.....	12
Aufbau einer dauerhaft leuchtenden LED.....	13
Aufbau einer gesteuerten LED (Blinken).....	14
Modifizieren.....	14
Aufbau eines Helligkeitssensors.....	15
Evaluation.....	16
Fazit.....	17
Bildquellen.....	18

Herleitung

Der Lehrplan Gymnasium Informatik ([LPInf]) bietet mehrere Schnittstellen zur Integration des Arduino Uno in den Unterricht in Fach Informatik:

- Klasse 7, LB1 Computer verstehen
 - EVA-Prinzip
- Klasse 9/10, LB4 Algorithmen und Programme
 - Einfache Datentypen
 - Algorithmische Grundstrukturen
 - Modularisierung
- Klasse 9/10, WP1: Messen, Steuern, Regeln
 - Einfache Steuerungen, Sensoren und Aktoren
- Klasse 11/12, LB 8B: Technische Informatik
 - Zusätzlich: Regelkreise, Signalwandler, Schnittstellen

Gegenüber anderen Werkzeugen zum Erlernen der Grundlagen der Programmierung, hat der Arduino Uno einen nennenswerten Bezug zum Aufbau elektronischer Schaltungen. Dies legt es nahe, den Arduino Uno in den Lernbereichen WP1 (Messen, Steuern, Regeln) in der Klassenstufe 9/10 oder LB 8B (Technische Informatik) in der Klassenstufe 11/12 einzusetzen.

Aus dem gleichen Grund lohnt auch ein Blick zur Physik ([LPPhy]):

- Klasse 6, LB4: Elektrische Stromkreise
- Klasse 7, LB2: Stromstärke und Spannung, WP2: Elektrische Schaltungen
- Klasse 8, LB3: Elektrischer Widerstand
- Klasse 9, LB1: Grundlagen der Elektronik

Danach kann erwartet werden, dass die Schüler die grundlegenden Kompetenzen für die Beschaltung des Arduino Uno mit Aktoren und Sensoren besitzen bzw. entsprechende Handreichungen (Schaltplan, Abbildung) verwenden können.

Einordnung

Andere Werkzeuge

Für den Inhaltsbereich Algorithmen werden in den fortgeschrittenen Jahrgängen **Editoren** und **Integrierte Entwicklungsumgebungen** (IDE) für textuelle Programmiersprachen benutzt. Der Übergang ist fließend. Professionelle IDE, wie z.B. Eclipse (Java u.a.) oder IntelliJ IDEA (Java u.a.) sind wegen des Funktionsumfangs im Unterricht eher ungeeignet. Einfachere IDE wie Thonny (Python) oder Code::Blocks (C, C++ u.a.) sind da angemessener. Letztere ist auch bereits mit der Arduino IDE vergleichbar. Die Arduino IDE unterscheidet sich vor allem dadurch, dass sie speziell zur Erstellung von Programmen für die Arduino Hardware Plattform ausgelegt ist.

Zum Einstieg in die Erstellung von Algorithmen existieren **grafische Programmierungsumgebungen**. Häufig sind die Anweisungen auf gut vorstellbare Aktionen von Sprite-Grafiken o.ä. fokussiert. Beispiele sind Karel, Kara oder Scratch.

Werden Programme für die Ausführung auf speziellen Hardware-Umgebungen mit Aktoren und Sensoren erstellt, spricht man oft von **Physical Computing**. Hier wird auch der Arduino Uno eingeordnet obwohl die an die Programmiersprache C angelehnte Arduino IDE auch einen Einstieg in die textuelle Programmierung von Programmen auf reinen Computern bietet. Die folgenden beide Absätze vergleichen den Arduino Uno mit zwei verbreiteten Werkzeugen aus diesem Bereich.



Der **Calliope mini** hat auf einer Platine eine Vielzahl von Aktoren und Sensoren fest verbaut. Damit können ähnliche Projekte auch ohne Schaltungsaufbau realisiert werden. Damit ist der Betreuungsaufwand und die Einstiegshürde deutlich geringer. Die Programmierung kann in einem Webbrowser erfolgen. Auf <http://calliope.cc> stehen verschiedene Editoren zur Verfügung. Der Editor Makecode erlaubt wahlweise grafische und textuelle Programmierung. Die

kompilierten Programmdateien können heruntergeladen und in das USB-Laufwerk, als das der Calliope mini erkannt wird, kopiert werden.



Der BBC micro:bit kann als kleiner Bruder (bezüglich Preis und Ausstattung) des Calliope mini angesehen werden. Auch hier sind auf einer Platine eine Vielzahl von Aktoren und Sensoren fest verbaut. Die Programmierung kann in einem Webbrowser erfolgen. Auf <https://www.microbit.org/de/code/> stehen der Editor Makecode (grafische und textuelle Programmierung in JavaScript) und ein Editor für textuelle Programmierung in Python zur Verfügung. Die kompilierten Programmdateien können

heruntergeladen und in das USB-Laufwerk, als das der micro:bit erkannt wird, kopiert werden.

Vorhandene Lehr-Lern-Materialien

Aufgrund der Verbreitung des Arduino Uno existieren mehrere Sammlungen auch frei verfügbarer Projektbeschreibungen für den Arduino Uno. Diese haben teilweise bereits Bezug auf den Einsatz im Bildungsbereich.

Das Schülerlabor Informatik der RWTH Aachen stellt unter dem Titel "Informatik Enlightened" (<https://schuelerlabor.informatik.rwth-aachen.de/modulmaterialien/informatik-enlightened>) Materialien für mehrere Stationen experimentellen Lernens mit dem Arduino Uno zur Verfügung. Die Stationen sind (in Klammern die verwendeten Sensoren und Aktoren):

- Einstieg: Ansteuerung einer LED (Taster, LED), Nutzung der seriellen Konsole
- Sonnenblume: Nachführen einer Sonnenblume zu einer Lichtquelle (Helligkeitssensor, Servomotor)
- Einparkhilfe: Akustische Abstandswarnung (Abstandsensor, Piezo-Signalgeber)
- Geschwindigkeitsmessung: Lichtschranken (IR-Diode, IR-Sensor)
- Farbthermometer: farbliche Temperaturanzeige (RGB-LED, Temperatursensor)

Die Materialien sind jeweils eine Arbeitsanleitung mit der der Schüler selbstständig arbeiten kann. Vorversuche und Vorüberlegungen führen auf die verwendeten Bauteile und Programmkonstrukte hin. Die Aufbauten sind ausführlich erklärt und abgebildet.

Betreuung ist zumindest für die Beantwortung von Verständnisfragen und bei der Fehlersuche erforderlich.

Motivation

Der Einsatz des Arduino Uno als digitales Lernwerkzeug im Unterricht ermöglicht die Vermittlung verschiedener Grundideen der Informatik:

- Die Unterteilung von Hardware und Software in die Funktionsbereiche Eingabe, Verarbeitung und Ausgabe (EVA-Modell): Durch die natürlich gegebene Zuordnung der Bauteile (Aktoren vs. Sensoren), Pins und verwendeten Befehle (`*read()` vs. `*write()`) ist diese Unterteilung auf verschiedenen Ebenen erfahrbar. Sofern die Modularisierung von Programmen thematisiert wird, kann durch entsprechende Funktionen für die Eingabe, Verarbeitung und Ausgabe das EVA-Modell auch auf dieser Ebene angewendet werden.
- Datentypen und Modularisierung: Die Programmiersprache und Programmstruktur der Arduino IDE verwendet getypte Variablen und bietet unmittelbar die Möglichkeit Bibliotheksfunktionen zu nutzen oder eigene Funktionen zu definieren.
- Algorithmische Grundstrukturen: Alternativen (if-then-else) werden auch in einfachsten Projekten mit Steuerungen benutzt. Der Arduino Uno kann hier als Einstieg dienen, der m.E. mehr Einsicht in die Wirkung von Alternativen vermittelt als z.B. das Anfertigen von Struktogrammen. Schleifenkonstrukte tauchen dagegen in einfachen Projekten oft nicht auf, da hier die vorgegebene `loop()`-Funktion implizit die Wiederholungen übernimmt. Hier müsste etwas mehr Zeit verwendet werden, um zu Programmen zu gelangen, die Schleifenkonstrukte sinnvoll einsetzen.
- Interaktion mit der Umgebung: Sensoren und Aktoren und ihre Einbindung in Programme und Steuerungen sind die Grundelemente von Arduino-Projekten. Der Arduino Uno eignet sich daher hier besonders und kann auf allen Schwierigkeitsstufen eingesetzt werden. Durch den universellen Aufbau (Experimentierplatine, universelle analoge und digitale Pins) steht eine praktisch unbegrenzte Auswahl zur Verfügung.

Diesen umfangreichen Möglichkeiten stehen verschiedenen Hürden gegenüber: Die erforderlichen Fertigkeiten beim Schaltungsaufbau und die textuelle Programmierung in einer an C angelehnten Sprache. Beim Einsatz im Unterricht ist also eine eher große Mindestzeit der Beschäftigung mit dem Arduino Uno, eine umfangreiche Handreichung für die Lernenden (z.B. Abbildungen) und eine ausreichende Betreuung (z.B. zur Fehlersuche) erforderlich.

Reihenplanung

Die erarbeitete Reihenplanung besteht aus 3 Doppelstunden für die Klassenstufe 9/10 an einem Gymnasium. Es wird hier die Bereiche

- Wahlpflicht 1: Messen, Steuern, Regeln mit Informatiksystemen und
- Lernbereich 4: Algorithmen und Programme

verbunden. Der Bereich WP1 wird dabei vollständig abgedeckt. Der LB4 wird nur zu einem geringen Teil abgedeckt, sowohl von der Stundenzahl als auch inhaltlich. Denkbar ist, die geplante Reihe als Einstieg in den LB4 oder als Wiederholung und Festigung des LB4 zu nutzen. Die sonstige Vermittlung im LB4 kann z.B. anhand einer grafischen Programmierumgebung wie Scratch erfolgen.

Der Arduino kombiniert Aspekte der Programmerstellung und des Schaltungsaufbaus auf einer Experimentierplatine. Das führt erfahrungsgemäß zu verschiedenen Problemen und Fehlern bis ein Aufbau läuft. Um die Schüler bei der Fehlersuche besser unterstützen zu können, wurde als Sozialform für die meisten Unterrichtsabschnitte Partnerarbeit gewählt. Zusätzlich scheint es sinnvoll, ein Hilfesystem der Schüler untereinander zu etablieren.

In der zweiten und dritten Doppelstunde werden mehrere Serien kleinerer gleichartiger Aufgaben (verschiedene Aktoren, verschiedene Sensoren, verschiedene Steuerungsszenarien) gestellt. Hier bietet sich durch verschiedene Aufgabenzahl bzw. Komplexität eine Differenzierung an, ohne dass bei schwächeren Schülern das grundlegende Lernziel verfehlt wird.

Stunde1: Grundlagen eines Aufbaus mit dem Arduino

Kurs: Klassenstufe 9/10	Teillernziele: Die Schüler (m/w/d) <ul style="list-style-type: none"> • bauen anhand von Schaltplänen und Abbildungen Schaltungen mit einem Sensor oder Aktor auf. • laden Programme in den Arduino und testen sie • vollziehen ein Programm zum Ansteuern eines Aktors (LED an/aus) nach und führen einfache Modifizierungen aus • vollziehen ein Programm zum Auswerten eines Sensors durch Ausgabe auf die serielle Konsole nach
Schulstunde in der Reihe: Programme und Steuerungen	
Thema der Stunde: Grundlagen eines Aufbaus mit dem Arduino	
Lernziel: Die Schüler (m/w/d) bauen einfache Schaltungen mit dem Arduino auf und testen sie.	

Zeit	Phase	Inhalt	Sozialform	Medien	Materialien	Vorwissen	Intendierter Lernprozess
15'	Motivation	Demonstration eines Aufbaus aus Schaltung und Programm	LV		Fertiger Aufbau, Programm	Grundbegriff Algorithmus	Einblick in die grundlegenden Elemente und Möglichkeiten eines Aufbaus
15'	Aufbau Einstieg 1	Aufbau einer dauerhaft leuchtenden LED	PA	Aufgabenstellung, Abbildung und Schaltplan (Ausdruck)	LED, Kabel	Stromkreis	Kennelernen der Bauteile, Fertigkeiten beim Aufbau
15'	Einstieg 2	Aufbau einer gesteuerten LED (Blinken)	PA	Aufgabenstellung, Abbildung, Schaltplan, Programm (Ausdruck)	LED, Kabel	Grundfertigkeit Computernutzung (z.B. Kopieren von Dateien)	Fertigkeiten beim Laden eines Programms, Einblick in die Struktur eines Programms und in die Grundschaltung eines Aktors
15'	Festigung	Modifizieren (Blinkgeschwindigkeit, Pausenverhältnis, Ausgabe eines Morsezeichens, Ändern des Pins)	PA	Aufgabenstellung			Einblick in die Wirkung der vorgegebenen Programmelemente (Ausgabe auf einen digitalen Pin)

20'	Erarbeitung	Aufbau eines Helligkeitssensors, Aufzeichnen der Abhängkt. Helligkeit - Sensorwert	PA	Aufgabenstellung, Abbildung, Schaltplan, Programm (Ausdruck, Arbeitsblatt)	Helligkeitssensor	Spannungsteiler	Einblick in die Grundschtaltung eines Sensors und die Programmelemente (Einlesen eines analogen Pins und Ausgabe auf die serielle Konsole
5'	Abbau	Abbau der Arbeitsmittel	PA				
5'	Rückblick	Nennen der grundlegenden Schritte beim Aufbau eines Aktors oder Sensors	UG				

Stunde2: Arbeit mit verschiedenen Aktoren und Sensoren

Kurs: Klassenstufe 9/10	Teillernziele: Die Schüler (m/w/d) <ul style="list-style-type: none"> • bauen anhand von Schaltplänen und Abbildungen Schaltungen mit verschiedenen Aktoren und Sensoren auf. • kombinieren einen Sensor und Aktor in einer Schaltung, laden und testen ein Programm und führen einfache Modifizierungen aus • benutzen if-Anweisungen im Programm
Schulstunde in der Reihe: Programme und Steuerungen	
Thema der Stunde: Arbeit mit verschiedenen Aktoren und Sensoren	
Lernziel: Die Schüler (m/w/d) bauen die Grundschtaltungen für verschiedene Aktoren und Sensoren und testen sie.	

Zeit	Phase	Inhalt	Sozial form	Medien	Materialien	Vorwissen	Intendierter Lernprozess
10'	Motivation	Demonstration eines Aufbaus aus Schaltung und Programm	LV		Fertiger Aufbau, Programm		Einblick in verschiedene Sensoren und Aktoren
25'	Erarbeitung 1	Aufbau und Ansteuerung verschiedener Aktoren (LED, RGB-LED, Piezo, Motor)	PA	Aufgabenstellung, Abbildung und Schaltplan (Ausdruck)	Aktoren, Kabel		Kennelernen der Aktoren und Erkennen des wiederkehrenden Aufbau- und Programmprinzips
25'	Erarbeitung 2	Aufbau und Ansteuerung verschiedener Sensoren (Helligkeit, Temperatur, Abstand)	PA	Aufgabenstellung, Abbildung und Schaltplan (Ausdruck)	Sensoren, Kabel		Kennelernen der Sensoren und Erkennen des wiederkehrenden Aufbau- und Programmprinzips

20'	Erarbeitung 3	Aufbau einer Steuerung mit Helligkeitssensor und LED, Modifizieren des Programms (Ändern der Schwelle zum Ein- und Ausschalten der LED)	PA	Aufgabenstellung, Schaltplan, Programm (Ausdruck)	LED, Helligkeitssensor		Einblick in die Programmstruktur einer Steuerung
5'	Abbau	Abbau der Arbeitsmittel	PA				
5'	Rückblick	Nennen der grundlegenden Schritte beim Aufbau mit Aktors und Sensors	UG				

Stunde3: Arbeit mit verschiedenen Programmelementen

Kurs: Klassenstufe 9/10	Teillernziele: Die Schüler (m/w/d) <ul style="list-style-type: none"> • modifizieren und erstellen Programme, um ein vorgegebenes Steuerverhalten zu realisieren • nutzen verschiedene Kontrollstrukturen und Modularisierungen bei der Programmerstellung
Schulstunde in der Reihe: Programme und Steuerungen	
Thema der Stunde: Arbeit mit verschiedenen Programmelementen	
Lernziel: Die Schüler (m/w/d) erweitern das Programm einer Steuerung unter Nutzung verschiedener Kontrollstrukturen und einfacher Modularisierungen.	

Zeit	Phase	Inhalt	Sozialform	Medien	Materialien	Vorwissen	Intendierter Lernprozess
10'	Motivation	Demonstration eines Aufbaus aus Schaltung und Programm	LV		Fertiger Aufbau, Programm		Einblick in Möglichkeiten, das Steuerungsverhalten zu verbessern

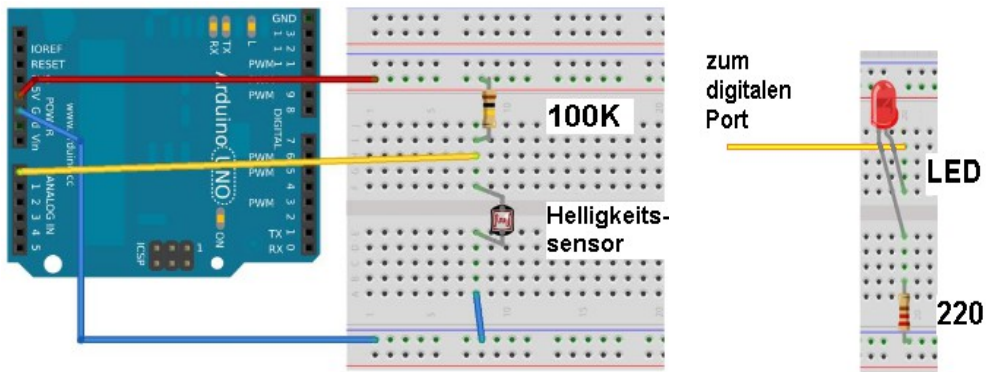
65'	Erarbeitung	Aufbau einer Steuerung mit Helligkeitssensor und LED, Testen, Modifizieren bzw. Erstellen eines Programms zur Änderung des Steuerungsverhaltens	PA	Aufgabenstellung, Schaltplan und Programme (Ausdruck)	LED, Helligkeitssensor		Kennelernen grundlegender Kontrollstrukturen und Modularisierungen in Programmen
5'	Abbau	Abbau der Arbeitsmittel	PA				
10'	Rückblick	Nennen und Beschreiben von möglichen Steuerungen, die mit dem Arduino aufgebaut werden können	UG				

Lehr-Lern-Material

Die folgenden Abschnitte enthalten die Materialien für die Doppelstunde "Grundlagen eines Aufbaus mit dem Arduino". Die Abschnitte entsprechen den oben geplanten Unterrichtsabschnitten.

Demonstration eines Aufbaus aus Schaltung und Programm

Als Aufbau wurde eine Beleuchtungssteuerung mittels Helligkeitssensor ausgewählt. Die Schaltung wird vorher fertig aufgebaut (zur Übersichtlichkeit der Verdrahtung ist die LED abgesetzt dargestellt):



Folgende Komponenten werden gezeigt und genannt:

- Arduino Board, Anschluss des Boards an den Computer, Experimentierplatine
- LED und Helligkeitssensor
- Pins am Board: Spannung GND und 5V, analoge und digitale Ein- bzw. Ausgänge

Das Programm wird ohne Erläuterung des Inhalts gezeigt, auf den Arduino Uno eingespielt und die Funktion getestet.

```
int x;

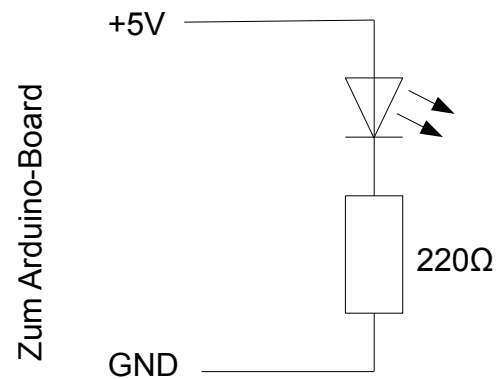
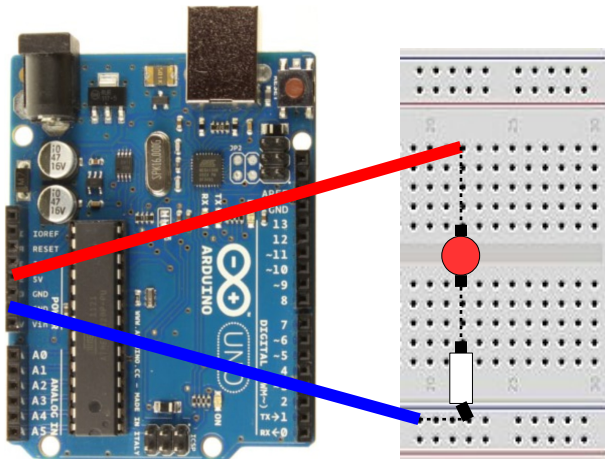
void setup() {
  pinMode(2, OUTPUT);
}

void loop() {
  delay(500);
  x = analogRead(0);
  // Zahl 300 anpassen
  if (x > 300) {
    digitalWrite(2, HIGH)
  } else {
    digitalWrite(2, LOW)
  }
}
```

Aufbau einer dauerhaft leuchtenden LED

In diesem Unterrichtsabschnitt sollen die Schüler den Anschluss des Arduino Uno, den Umgang mit der Experimentierplatine, Bauteilen und Kabeln erlernen. Anhand dieses sehr einfachen Stromkreises wird der Zusammenhang Schaltplan – Abbildung – Schaltung erfahren.

Aufgabenstellung: Baue die Schaltung gemäß Schaltplan und der Abbildung auf. Achte darauf, dass ein geschlossener Stromkreis vom Pin 5V über die Leuchtdiode und den Widerstand zum Pin GND entsteht.

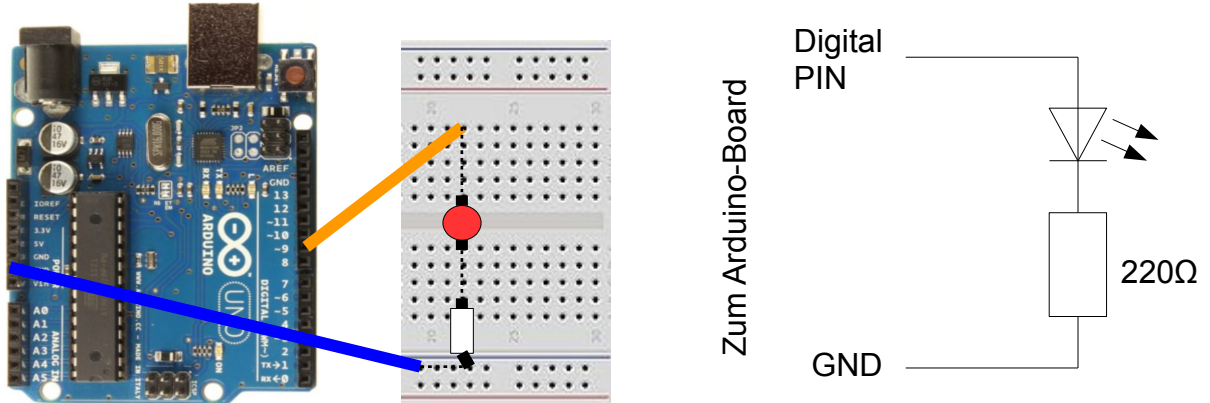


Aufbau einer gesteuerten LED (Blinken)

In diesem Unterrichtsabschnitt sollen die Schüler den Prozess der Programmerstellung durchlaufen.

Aufgabenstellung: In diesem Experiment soll die LED nicht mehr dauerhaft leuchten, sondern durch ein Programm gesteuert blinken.

1. Ändere den Aufbau gemäß Schaltplan und der Abbildung. Das obere Ende der Leuchtdiode wird jetzt mit einem digitalen Pin verbunden. Zunächst wird sie nicht leuchten.



2. Starte die Arduino Software auf dem PC. Gib das abgebildete Programm ein. Die // und der Text dahinter sind Kommentare und müssen nicht mit eingegeben werden.

```
void setup() {                                // wird am Anfang ausgeführt
  pinMode(9, OUTPUT);                          // der digitale Pin 9 soll Ausgabe sein
                                              // (LED ist mit Pin 9 verbunden)
}

void loop() {                                  // wird immer wiederholt
  delay(500);                                  // 0,5s warten, damit die LED nicht
                                              // zu schnell blinkt
  digitalWrite(9, HIGH);                       // LED anschalten - Spannung an Pin 9
  delay(500);                                  // 0,5s warten
  digitalWrite(9, LOW);                        // LED ausschalten
}
```

3. Lade das Programm in den Arduino Uno (Knopf ). Die LED soll blinken.

Modifizieren

Aufgabenstellung: Nimm am Programm verschiedene Änderungen vor und teste.

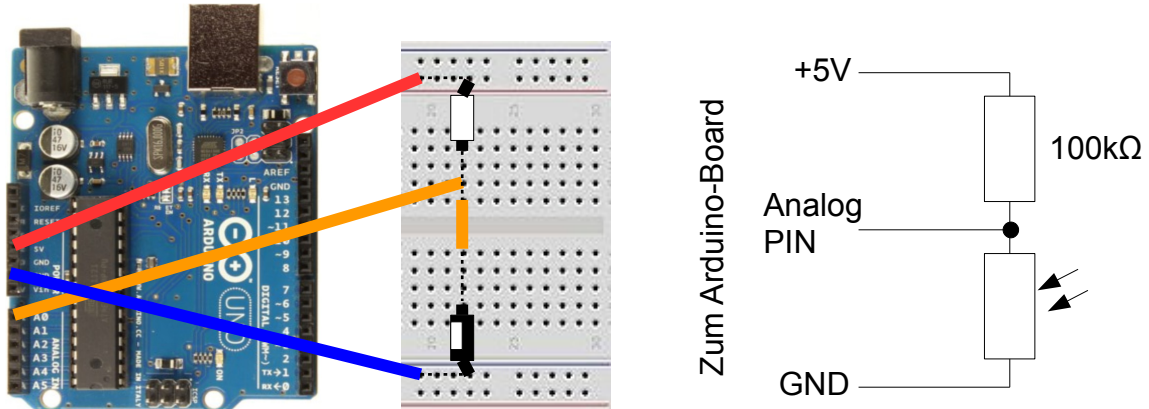
- Ändere die Blinkgeschwindigkeit (Wartezeiten delay ändern)
- Ändere die Wartezeiten so, dass die LED immer nur kurz aufblitzt
- die LED das Morsezeichen C (lang-kurz-lang-kurz) ausgibt
- der digitale Pin 7 statt 9 benutzt wird (Umstecken an Schaltung und Programm ändern)

Aufbau eines Helligkeitssensors

In diesem Unterrichtsabschnitt lernen die Schüler den Anschluss eines Sensors und die Auswertung der Sensorwerte im Programm kennen.

Aufgabenstellung:

1. Baue die Schaltung gemäß Schaltplan und der Abbildung auf. Achte darauf, dass ein geschlossener Stromkreis vom Pin 5V über den Widerstand und den Helligkeitssensor zum Pin GND entsteht. Aus diesem Stromkreis zweigt das gelbe Kabel zum analogen Pin A0 ab.




2. Starte die Arduino Software auf dem PC. Gib das abgebildete Programm ein. Die Kommentare und müssen wieder nicht mit eingegeben werden.

```
int x; // Variable um Wert zu speichern

void setup() {
  Serial.begin(9600); // serielle Konsole initialisieren
}

void loop() {
  delay(500); // 0,5s zwischen 2 Messungen warten
  x = analogRead(0); // Spannung am Pin A0 messen, in x speichern
  Serial.println(x); // .. und über die serielle Konsole anzeigen
}
```

3. Lade das Programm in den Arduino Uno (Knopf ).
4. Starte in der Arduino Software den Seriellen Monitor (Menü Werkzeuge). Im Monitor sollen forlaufend Zahlen erscheinen, Das sind die vom Programm gemessenen Spannungen.
5. Verdunkle mit der Hand den Helligkeitssensor in der Schaltung unterschiedlich stark und beobachte die Veränderung der ausgegebenen Zahl. Fertige eine Tabelle an, die den Zusammenhang der Verdunklung zu den ausgegebenen Zahlen darstellt.

Evaluation

Im Seminar traten vor allem folgende Schwierigkeiten auf:

- Die Teilnehmer hatten Schwierigkeiten mit dem Schaltungsaufbau auf der Experimentierplatine. Zum einen wegen des engen Lochabstandes als auch der nicht aufgedruckten Verbindung der Löcher. Auch das Vorwissen zu Stromkreisen war sehr unterschiedlich.
- Die Teilnehmer hatten wenig Vorwissen zur Syntax der Programmiersprache.
- Die Fehlersuche im Schaltungsaufbau und im Programm verursachte hohen Betreuungsaufwand.

Es stellte sich als Problem heraus, dass mehrere neue Fertigkeiten (Schaltungsaufbau, Programm und die Bedienung der Arduino IDE) gleichzeitig benötigt wurden.

Positiv wurde bemerkt, dass durch den universellen Aufbau eine große Vielfalt von Aktoren und Sensoren und damit Steuerungsszenarien möglich ist. Das erschließt z.B. spezielle Anforderungen an Berufsschulen mit Lerninhalten aus dem Bereich der Elektronik.

Insgesamt wurde, vor allem aufgrund der genannten Schwierigkeiten, der Einsatz im Unterricht skeptisch beurteilt. Alternative Plattformen, z.B. Calliope mini, wurden als geeigneter angesehen.

Fazit

Ein Einsatz im regulären Unterricht erscheint mir nur angebracht, wenn der zusätzliche Aufwand des Schaltungsaufbaus durch entsprechende Lerninhalte (z.B. auch fächerübergreifend mit Physik) gerechtfertigt ist.

Der Einsatz im Rahmen einer GTA, Begabungsförderung oder Projektwoche ist jedenfalls möglich.

Die angesprochenen Probleme wurden in der Unterrichtsplanung in verschiedener Weise berücksichtigt:

- Einsatz von Partnerarbeit zur effizienteren Hilfestellung durch den Lehrer
- Erstes Experiment zum Schaltungsaufbau und zur Programmerstellung getrennt
- Verschiedene Darstellungen in den Aufgaben zum Schaltungsaufbau (Abbildung, Schaltplan)

Eine weitere Möglichkeit, die Einsatzmöglichkeiten zu erweitern, ist der Aufbau der Schaltungen mittels Bauteilen auf separaten Grundplatten mit Anschlusspins und Kabelverbindungen. Dadurch sind die Leitungswege besser sichtbar und dies ist oft den Schülern aus dem Physikunterricht bekannt.

Literaturverzeichnis

LPInf: Sächsisches Staatsministerium für Kultus, Lehrplan Gymnasium Informatik, 2018

LPPhy: Sächsisches Staatsministerium für Kultus und Sport, Lehrplan Gymnasium Physik, 2011

Bildquellen

Abschnitt Einordnung

https://commons.wikimedia.org/wiki/File:BBC_micro_bit.jpg, Les Pounder from Blackpool, UK
[CC BY-SA 2.0 (<https://creativecommons.org/licenses/by-sa/2.0>)]

https://commons.wikimedia.org/wiki/File:Calliope_mini_weiss_JoernAlraun.jpg, Jørn Alraun [CC
BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0>)]

Abschnitt Lehr-Lern-Material

<https://commons.wikimedia.org/wiki/File:Arduino-uno.jpg>,
Imik tech [CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0>)], bearbeitet

<https://commons.wikimedia.org/wiki/File:Fritzing-steckplatine.jpg>,
fritzing.org [CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0>)], bearbeitet