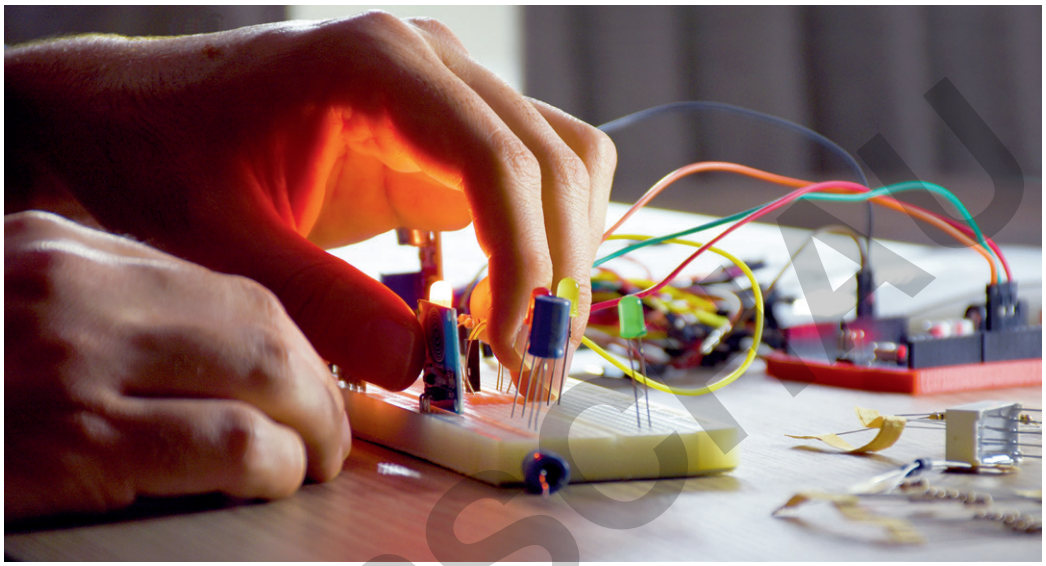
III.29

Natur und Technik

Mit dem Mikrocontroller Arduino zur Ampelschaltung – Flüssigen Verkehr programmieren

Nach einer Idee von Thomas Rosenthal
Mit Illustrationen von Dr. Wolfgang Zettlmeier



© s-cphoto/E+

Im Verkehr spielt Energieverbrauch eine große Rolle. Ampeln sind dazu da, einen flüssigen Verkehr zu ermöglichen und dadurch unter anderem die Energieeffizienz zu steigern. Vermitteln Sie mithilfe dieses Beitrags ein Grundverständnis der Funktionsweise von Ampeln und befähigen Sie so die Lernenden, die Technik hinter einer Ampelschaltung zu begreifen. Anhand des Mikrocontrollers Arduino lernt Ihre Klasse selbstständig einzelne Bauteile zu programmieren, bei einer Autofahrerampel eine Tag-und-Nacht-Situation zu simulieren sowie bei der Fußgängerampel per Taster die Farbe Grün, begleitet von einem Ton, anzufordern.

KOMPETENZPROFIL

Klassenstufe:	8–10
Dauer:	10 Unterrichtsstunden
Kompetenzen:	Die Lernenden … 1. beschreiben Aufbau und Funktionsweise von Mikrocontrollern, 2. programmieren einzelne Bauteile und führen sie in einem Programm zusammen.
Thematische Bereiche:	Mikrocontroller, Arduino, Ampelschaltung, EVA-Prinzip, Programm, Softwareumgebung
Zusatzmaterialien:	Schaltpläne, Sketches, PowerPoint mit Infomaterial

Auf einen Blick

Ab = Arbeitsblatt, **Üs** = Übersichtsblatt

1./2. Stunde

Thema: Mikrocontroller im Alltag

M 1 (Ab) **Von Mikrocontrollern umgeben**
M 2 (Ab) **Der Mikrocontroller im Autoschlüssel**

3.–10. Stunde

Thema: Programmieren mit dem Mikrocontroller Arduino

M 3 (Ab) **Das Arduino-Elektronik-Hardware-Board**
M 4 (Ab) **Die Arduino-Software zum Programmieren des Boards**
M 5 (Ab) **Mit Arduino Lichter leuchten lassen**
M 6 (Ab) **Autofahrerampel – Steckplatinen nutzen**
M 7 (Ab) **Fußgängerampel – Variablen einsetzen**
M 8 (Ab) **Audioelemente einbinden**
M 9 (Ab) **Autofahrerampel – Tag-und-Nacht-Situation simulieren**
M 10 (Ab) **Mit einem Taster die Ampelphasen steuern**
M 11 (Üs) **Glossar zu den Arduino-Befehlen**

Benötigte Materialien: Pro Person:
- ☐ 1 Laptop/PC
- ☐ 1 USB-Stick
- ☐ 1 Arduino-Starterkit

Benötigte Dateien:
- ☐ Infomaterial.pptx
- ☐ Sketches.zip
- ☐ Schaltplaene.zip

Von Mikrocontrollern umgeben M 1

Aufgaben

1. **Betrachte** die folgenden Abbildungen und **beschreibe** Funktionen der einzelnen abgebildeten Geräte.
2. **Notiere** mögliche technische Gemeinsamkeiten.
3. **Nenne** weitere Geräte, die du kennst und **beschreibe** deren Funktionsweise.

© von oben im Uhrzeigersinn: Thinkstock/iStock; ALAMA/iStock/Getty Images Plus; pictafolio/E+; mbbirdy/E+; scanrail/iStock/Getty Images Plus; alexsl7E+

Mit Arduino Lichter leuchten lassen

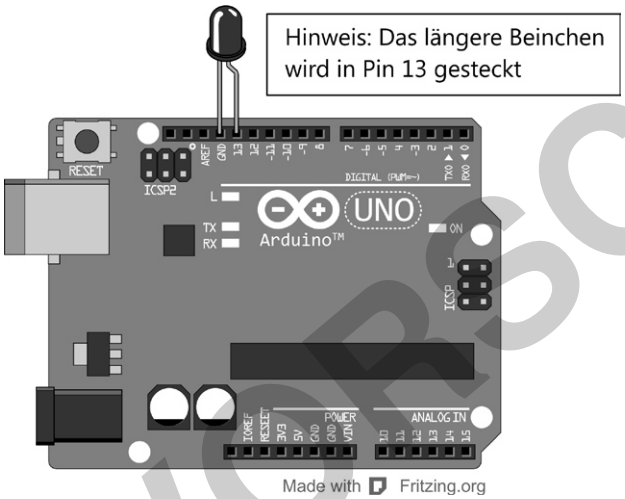
M 5

Nun kommen wir dazu, wie man den Arduino nutzen kann, um beispielsweise Lichter (LEDs) steuern zu können. Dazu können die insgesamt 20 Anschlussports (0–13 und A0 bis A5) des Arduino genutzt werden. Diese sind wie kleine anschaltbare Steckdosen. Bei jeder Steckdose ist immer ein Anschluss mit dem Potenzial 0 V vorhanden. Beim Arduino ist dies der Anschluss GND (Ground). Für den zweiten Anschluss mit dem hohen Potenzial lassen sich beim Arduino die Ports zwischen 5 V und 0 V an- und ausschalten. Das elektrische Potenzial ist so etwas wie ein „elektrischer Druck" auf einem Kabel. Ganz ähnlich kann auch in einer Wasserleitung ein hoher oder tiefer Wasserdruck herrschen. Verbindet man nun eine Leitung hohen Drucks mit einem Bereich, in dem ein tiefer Druck herrscht, dann fließt das Wasser. Genauso ist es auch beim Strom: Nur wenn es einen Potenzialunterschied gibt, können Elektronen fließen. Der Potenzialunterschied ist die bekannte Spannung.

Tipp

Sowohl das Potenzial als auch der Potenzialunterschied haben die Einheit Volt. Deshalb werden die beiden Begriffe Spannung und Potenzial oft verwechselt.

Mit der folgenden Schaltung kann eine LED zum Leuchten gebracht werden:



So geht's

Stecke die LED, wie in der obigen Abbildung beschrieben, schreibe den folgenden Sketch in der Arduino-Software und lade ihn anschließend auf den Arduino hoch. Was beobachtest du? Speichere den Sketch anschließend unter „**01_Blinkende_LED**".

© Arduino www.arduino.cc

Fußgängerampel – Variablen einsetzen

M 7

Variablen können ein umfangreiches Programm wesentlich vereinfachen und tragen dazu bei, bei vielen angeschlossenen Komponenten die Übersicht zu behalten. Verschiedenen Anschlüssen können über Variablen konkrete Namen zugeordnet werden, sodass im späteren Programm sehr einfach diese Namen verwendet und vor allem wiedererkannt werden können. Wenn man später mehrere Bauteile gleichzeitig einsetzt, behält man einen besseren Überblick, weil man keine Zahlen wie zum Beispiel die Ports des Arduinos, sondern lediglich die zuvor zugeordneten Namen der entsprechenden Bauteile verwendet.

Tipp

Bei der Namensgebung dürfen keine Umlaute oder Sonderzeichen verwendet werden.

Beispiel eines Programmes mit zwei LEDs:

Ohne Variablen

```
/*
Abwechselndes Blinken
Rote LED an Port 13, Grüne LED an
Port 12
*/

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  digitalWrite(12, LOW);
  delay(1000);
  digitalWrite(13, LOW);
  digitalWrite(12, HIGH);
  delay(1000);
}
```

Mit Variablen

```
/*
Abwechselndes Blinken
Rote LED an Port 13, Grüne LED an
Port 12
*/

int ledrot = 13;
int ledgruen = 12;

void setup()
{
  pinMode(ledrot, OUTPUT);
  pinMode(ledgruen, OUTPUT);
}

void loop()
{
  digitalWrite(ledrot, HIGH);
  digitalWrite(ledgruen, LOW);
  delay(1000);
  digitalWrite(ledrot, LOW);
  digitalWrite(ledgruen, HIGH);
  delay(1000);
}
```

Grafik: Dr. Wolfgang Zettlmeier

Aufgabe 1

Vergleiche den Sketch „Ohne Variablen“ mit dem Sketch „Mit Variablen“: **Notiere**, wie viele Befehle du jeweils umschreiben musst, wenn du die LEDs statt an den Ports 12 und 13 an den Ports 10 und 11 anschließen willst.

So geht’s

- Schreibe dein Sketch der Autofahrerampel so um, dass du für die drei LEDs Variablen verwendest. Notiere deine Überlegungen, welche weiteren Variablen du einführen könntest.
- Speichere den Sketch unter „**03_Autofahrer-Ampel_Variable**“.
- Erweitere dein Sketch der Autofahrerampel um eine Fußgängerampel. Beachte dabei den korrekten Ablauf der einzelnen Ampelfarben und gib sinnvolle Zeiten ein.

Speichere den Sketch unter „**04_Ampelkreuzung**“.

Aufgabe 2

Ergänze den *int*-Befehl in deinem Glossar.

M 8 Audioelemente einbinden

Mit dem Arduino kann man auch einzelne Töne und sogar ganze Melodien an einem kleinen Lautsprecher abspielen. Der Lausprecher wird dabei mit einem digitalen Port und GND verbunden. Der zugehörige Befehl beim Arduino heißt *tone*:

```
tone(Port, Frequenz, Dauer des Tones in ms);
```

Der Port, an den der Lautsprecher angeschlossen wird, muss nicht extra als OUTPUT definiert werden. Auch dies erledigt der *tone*-Befehl mit:

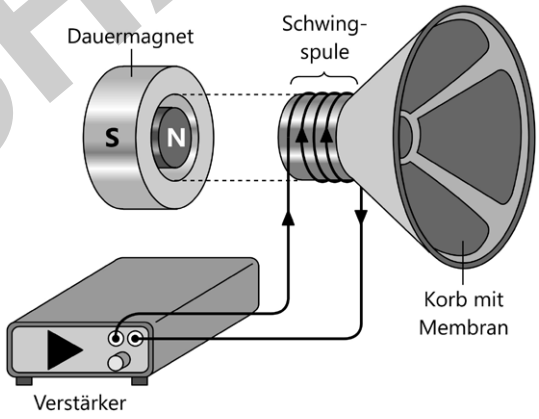
```
1    void loop()
2    {
3              tone(11,440,1000);
4              delay(2000);
5    }
```

Port 11 wird 440-mal pro Sekunde an- und ausgeschaltet und das für 1000 Millisekunden (1 Sekunde).

Sketch 1

Wie funktioniert ein Lautsprecher?

Lautsprecher bestehen aus einem Permanentmagnet (auch Dauermagnet genannt) und einer Spule, die als Elektromagnet fungiert. Sobald durch die Spule ein Strom fließt, entsteht ein Magnetfeld. Dreht man die Stromflussrichtung um, kehrt sich auch die Richtung des Magnetfeldes um. Je nach Richtung ihres Magnetfeldes wird die Spule vom Permanentmagnet entweder angezogen oder abgestoßen. Ändert man die Stromflussrichtung sehr schnell, schwingt die Spule ständig vor und zurück.



Grafik: Dr. Wolfgang Zettlmeier

Da die Spule mit einer dehnbaren Membran verbunden ist, wird die Luft vor dem Lautsprecher ebenfalls in Schwingung versetzt. Bei Schwingfrequenzen zwischen etwa 20 Hertz und 20 000 Hertz nehmen wir die Luftschwingungen mithilfe unserer Ohren als Ton wahr.

Da der Arduino nur Gleichstrom ausgeben kann, kann der Strom durch den Lautsprecher immer nur in eine Richtung fließen. Um trotzdem einen Ton zu erhalten, wird der Lautsprecher immer wieder ein- und ausgeschaltet. Weil die gedehnte Membran dann nicht mehr abgestoßen wird, bewegt sie sich von allein in ihre Ausgangsposition zurück. Wird der Stromfluss nun schnell hintereinander an- und abgeschaltet, bewegt sich die Membran ebenfalls schnell vor und zurück. Allerdings nicht ganz so stark, wie wenn man Wechselstrom verwenden würde. Der Ton ist deshalb nicht ganz so laut.