SINUS Nordrhein-Westfalen Informatik

Arduino – Auf dem Weg zum eigenen Projekt

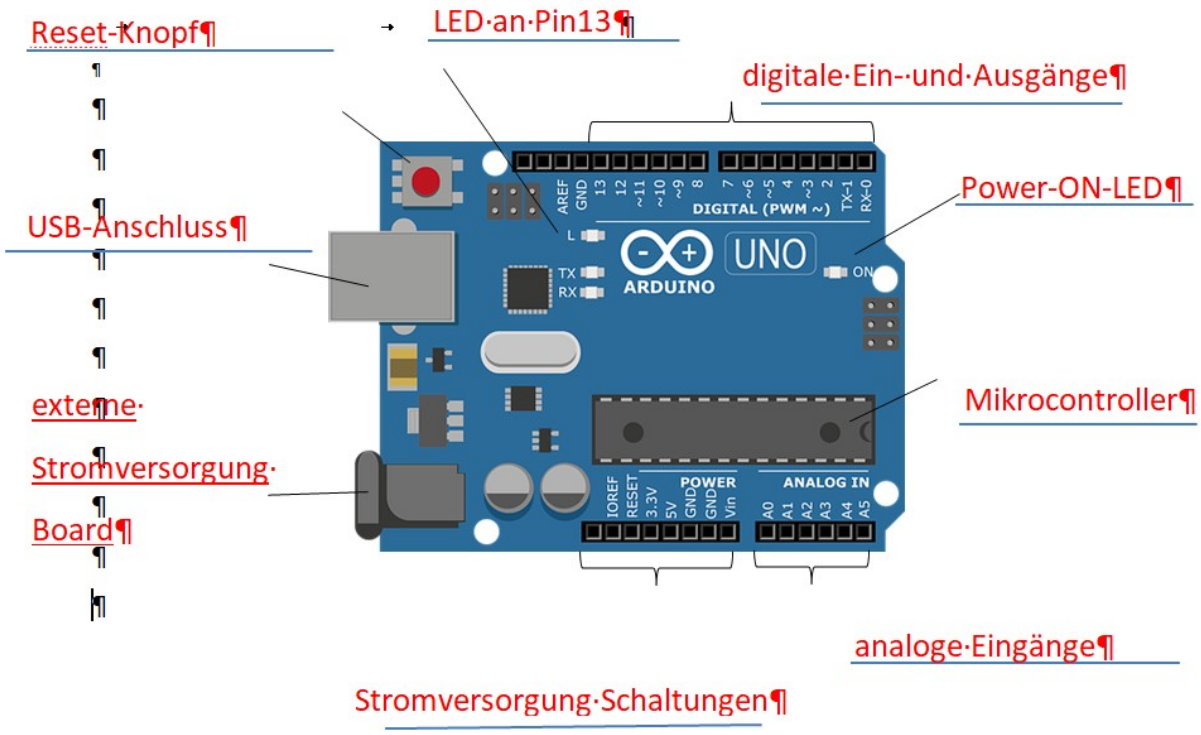
M0 Hinweise für die Lehrkraft

Die unten stehenden Hinweise dienen zusätzlich zur Erklärung für die Schülerinnen und Schüler. Das Bild des Funduino kann als stummer Impuls zum Stundeneinstieg genutzt werden. Zum besseren Verständnis sollten die Schülerinnen und Schüler ein Originalbauteil zur Verfügung gestellt bekommen.

Der „Arduino“ ist ein sogenanntes Mikrocontroller-Board. Das ist eine Leiterplatte (Board) mit einem Mikrocontroller und viel Elektronik. Am Rand des Boards befinden sich Steckplätze (Pins), an denen man die unterschiedlichsten Dinge wie zum Beispiel Schalter, LEDs, Ultraschallsensoren, Temperatursensoren, Drehregler, Displays, Motoren anschließen kann.

Es gibt verschiedene Versionen von Boards, die mit der Arduino-Software verwendet werden können. Dazu gehören sowohl viele verschiedene große und kleine Boards mit der offiziellen „Arduino“-Bezeichnung als auch viele günstigere, aber gleichwertige kompatible Boards. Typische offizielle Boards heißen Arduino UNO, Arduino MEGA, Arduino Mini… Kompatible Boards heißen Funduino UNO, Funduino MEGA, Freeduino, Seeduino, Sainsmart UNO usw.

Die Abbildung zeigt den Funduino UNO.



Wir haben uns für den Shop Funduinoshop.com entschieden und zwar für das Funduino Starter Lernset. Für dieses Lernset werden auf der Plattform www.funduino.de auch Lernsets angeboten. Außerdem kann hier auch ein Arbeitsheft zum Erlernen der Arduino

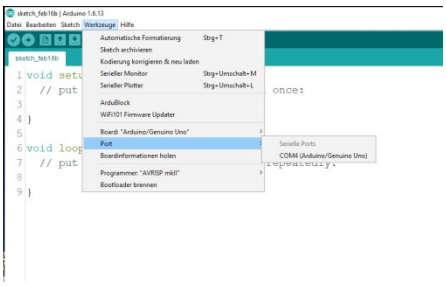
Entwicklungsumgebung erworben werden. Der Shop bietet für BIldungseinrichtungen ein Rabatt von 10%.

Die Software und das Programmieren

Die benötigte Software bekommt man kostenlos auf www.arduino.cc unter dem Menüpunkt Software – Download the Arduino IDE. Wenn man sich irgendwo aufgefordert fühlt, etwas zu bezahlen, handelt es sich nur um die Bitte nach einer freiwilligen Spende, der man nicht zwangsweise nachkommen muss. Für kostenlosen Download einfach „JUST DOWNLOAD“ wählen.

Man sucht dann das passende Betriebssystem aus und folgt den Anweisungen auf dem Bildschirm. Leider steht dort alles auf Englisch, aber die Installation sollte trotzdem zu bewältigen sein. Zu bedenken ist noch, dass man möglicherweise Administratorrechte haben muss, um die Software einschließlich Treiber installieren zu dürfen.

Es empfiehlt sich, vorsichtshalber alle vorgeschlagenen Treiber zu installieren und an jedem Arbeitsplatz mit einem Arduino auszuprobieren, ob er korrekt erkannt wird, bevor man mit den Schülerinnen und Schülern arbeitet. Die Schüler und Schülerinnen müssen immer darauf achten, dass unter dem Menüpunkt Werkzeuge sowohl das richtige Board als auch der richtige Port eingestellt sind. (siehe Abb.)



Da Software fortlaufend aktualisiert wird, können hier nur Tipps gegeben werden.

Bei der Programmiersprache für den Arduino handelt es sich genau genommen um C++. Das merkt man jedoch nur bei grundlegenden Befehlen wie Schleifen, bedingten Anweisungen und Verzweigungen sowie der Verarbeitung von Variablen. Die meisten anderen Befehle sind eigentlich sogenannte Methoden, die wie Befehle aussehen. Die Anweisung zur Konfiguration eines Portpins lautet beispielsweise *pinMode(7, OUTPUT);*
Pin 7 wird als Ausgang eingestellt.

Da viele wichtige Funktionen bereits verfügbar sind und zudem verständlich formuliert sind, muss man kein Informatikstudium mehr absolviert haben, um Programme – die beim Arduino Sketches heißen – zu entwickeln. Zugegeben: Ein paar Englischkenntnisse muss man schon noch mitbringen, aber die halten sich in Grenzen

Mit dieser „Arduino-Software“ schreibt man kleine Programme, sogenannte „Sketche", die der Mikrocontroller später ausführen soll.

Per USB-Kabel werden die fertigen Sketche auf den Mikrocontroller übertragen.

Jeder Sketch enthält mindestens die beiden Funktionen:

```
void setup() { }
void loop() { }
```

Die **setup**-Funktion wird zuerst ausgeführt. Sie wird genau **einmal** abgearbeitet.

Hier werden beispielsweise Hardwareeinstellungen vorgenommen. Man kann dort festlegen, ob ein Arduino-Anschluss ein Ausgang oder ein Eingang sein soll.

Wird ein Anschluss als **Ausgang** definiert, wird eine **Spannung am Board ausgegeben**. Beispiel: Mit diesem Pin soll eine Leuchtdiode zum Leuchten gebracht werden.

Wird ein Anschluss als **Eingang** definiert, soll vom Board eine **Spannung eingelesen** werden. Beispiel: Es wird ein Schalter gedrückt. Das Board bemerkt dies dadurch, dass es an diesem Eingangspin eine Spannung erkennt.

Die **loop**-Funktion wird **endlos** abgearbeitet.
Loop heißt auf deutsch: Schleife.
In diese Schleife kommen die Anweisungen, die der Arduino immer wieder ausführen soll.

Das sieht dann insgesamt so aus:

```
void setup()
{
Anweisungen;
}

void loop()
{
Anweisungen;
}
```

Anweisungen (auch: "Befehle") sagen dem Arduino, was er machen soll.
Jede Anweisung wird mit einem **Semikolon** abgeschlossen, z. B. **pinMode(13, OUTPUT);**

Weiterhin findet man häufig sogenannte Kommentare.

In Kommentare kann man Informationen oder Erklärungen zum Programm schreiben oder auch Programmzeilen nicht abarbeiten lassen.

Mehrzeilige Kommentare schließt man zwischen /* und */ ein.
Einzeilige Kommentare stehen hinter zwei Schrägstrichen, z.B.: **//500ms**

LEDs und Widerstände

Der genauere Aufbau einer LED wird nicht besprochen. Ebenso wird nicht auf die Berechnung der Widerstände eingegangen. Man stellt den Schülerinnen und Schüler entweder gleich nur die passenden Widerstände zur Verfügung oder lässt sie mit Hilfe einer Farbtabelle den benötigten Widerstand aussuchen. Genauere Informationen sollte an dieser Stelle der Physikunterricht liefern. Es wäre schön, wenn eine fächerübergreifende Arbeit möglich wäre.

Ob man 200 Ohm, 220 Ohm, 300 Ohm oder 330 Ohm Widerstände verwendet, macht keinen sichtbaren Unterschied. Ohne Widerstand jedoch ist die Leuchtdiode so schnell durchgebrannt, dass man kaum sieht, dass sie je an war. Verwendet man zu hohe Widerstände, bekommt die LED zu wenig Strom und leuchtet nicht, ist aber nicht kaputt.

Der häufigste Fehler der Schülerinnen und Schüler ist, dass Plus- und Minuspol vertauscht werden, obwohl auf die Wichtigkeit des richtigen Anschlusses hingewiesen wurde. Dann muss die Leuchtdiode einfach nur herumgedreht werden.

Die Farbe und Länge der verwendeten Kabel ist beliebig und hat keinen Einfluss auf die Funktionstüchtigkeit des Modells. Jedoch ist darauf zu achten, dass die Kabel richtig eingesteckt sind, das heißt, tief genug und im richtigen Pin.

Die Schaltpläne wurden mit www.fritzing.org erstellt. Auch diese Software kann man kostenlos downloaden, um eigene Schaltpläne zu erstellen.