Dimmen und Farben

Bisher kannst du OUTPUT-Pins HIGH oder LOW setzen und damit z.B. LEDs ein oder ausschalten. An allen Pins, die mit einer kleinen Tilde versehen sind (~) kann der Arduino eine LED aber auch dimmen[1] - zumindest kann er das simulieren.

Aufgabe 9.1

Schließe eine LED mit entsprechendem Schutzwiderstand an einen der geeigneten Pins an und probiere das Verändern der Helligkeit aus:

```
void setup(){
  pinMode(3,OUTPUT);
}
void loop(){
  digitalWrite(3,HIGH);
  delay(1000);
  digitalWrite(3,LOW);
  delay(1000);
  analogWrite(3,127);
  delay(1000);
}
```

Diese Zeilen machen das, was du schon kennst: high– und low-schalten von Pin 3.

Diese Zeile schaltet Pin 3 mit einer Stärke von 127 an. Die 127 steht für $\frac{127}{255}$, also ungefähr „halb an“. Bei 255 wäre die LED komplett hell, bei 0 wäre sie komplett dunkel.

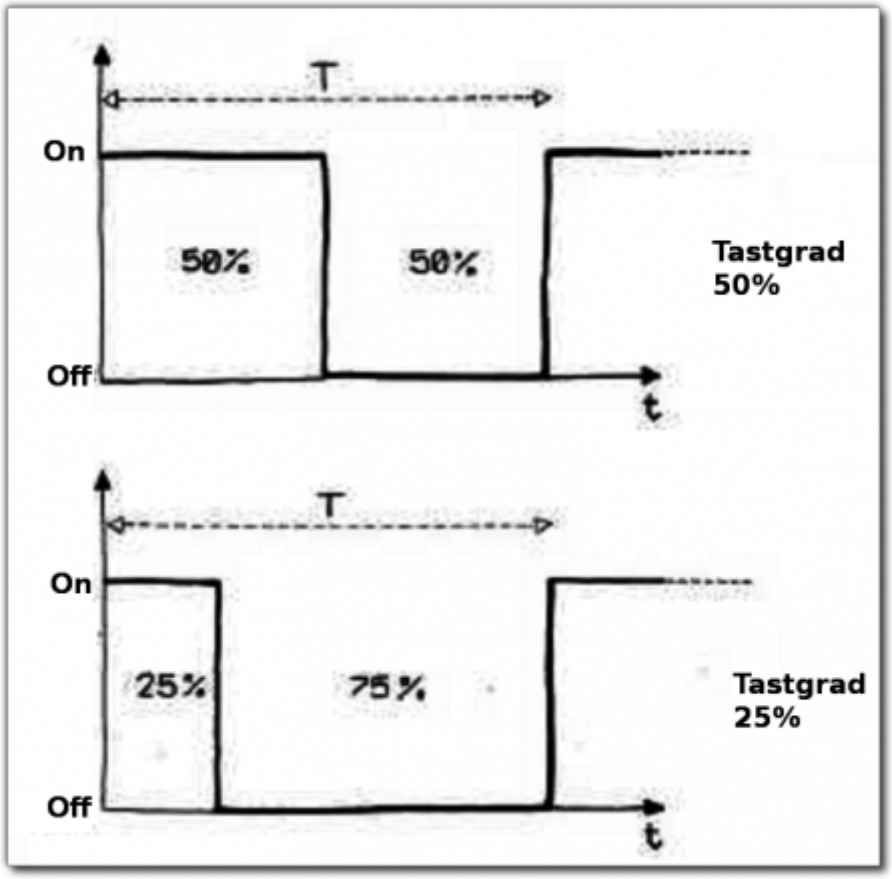
Informationen: PWM (PulsWeitenModulation)

Ein digitaler Pin kennt die beiden Zustände HIGH und LOW. Dann liegen an diesem Pin entweder 0V oder 5V an. Einige der digitalen Pins auf dem Arduino-Board sind mit „PWM“ (oder „~“) gekennzeichnet. Mit dem Befehl analogWrite(pin, value) kann man an diesen Pins eine pseudo-analoge Spannung erzeugen: **Analog ist das Gegenteil von digital.** Es bedeutet, dass Signale stufenlos oder zumindest fein gestuft verarbeitet werden.

Der Befehl analogWrite(11, 0) generiert eine gleichmäßige Spannung von 0 Volt an Pin11, der Befehl analogWrite(11, 255) generiert eine gleichmäßige Spannung von 5 Volt an Pin11. Für Werte zwischen 0 und 255 wechselt der Pin sehr schnell zwischen 0 und 5 Volt - je höher der Wert, desto länger ist der Pin HIGH (5 Volt).

Der Befehl analogWrite(11, 127) führt dazu, dass die Ausgangsspannung zur Hälfte der Zeit auf HIGH steht und zur anderen Hälfte auf LOW. Bei analogWrite(11, 192) 192 misst die Spannung am Pin zu einer Viertel der Zeit 0 Volt und zu Dreivierteln die vollen 5 Volt.

Weil dies eine hardwarebasierte Funktion ist, läuft die konstante Welle unabhängig vom Programm bis zur nächsten Änderung des Zustandes per analogWrite (bzw. einem Aufruf von digitalRead oder digitalWrite am selben Pin).



Hinweis: Die Anweisung `analogWrite()` funktioniert ideal mit den Pins 9 und10, bei 3 und 11 beeinflusst sie die `tone()`-Anweisung, bei Pins 5 und 6 geht eine LED nie ganz aus. Wie auch `tone()` schaltet `analogWrite()` einen Pin automatisch auf OUTPUT.

Aufgabe 9.2

Schreibe ein Programm, das eine LED immer langsam heller und dann wieder dunkler werden lässt. Zwei for-Schleifen helfen dir dabei.

Aufgabe 9.3

Setzt man in analog Write() die Werte 127 oder 128 ein, bekommt die LED ziemlich genau die halbe Energie. Trotzdem kommt das unserem Auge nicht „halb hell“ vor. Finde heraus, welcher Wert wirklich „halb hell“ aussieht. Haben deine Mitschülerinnen und Mitschüler ähnliche Werte?

Aufgabe 9.4

Unten siehst du ein Programm, das lauter zufällige Zahlen auf den seriellen Monitor schreibt. Mit `randomSeed (analogRead(A1))` wird der Zufallsgenerator einmal gestartet, mit `s=random (1,7);` immer eine Zufallszahl von 1 bis 6 erzeugt und in der Variablen s gespeichert. Kannst du das Programm so umbauen, dass du mit einer LED das zufällige Flackern einer Kerze simulieren kannst?

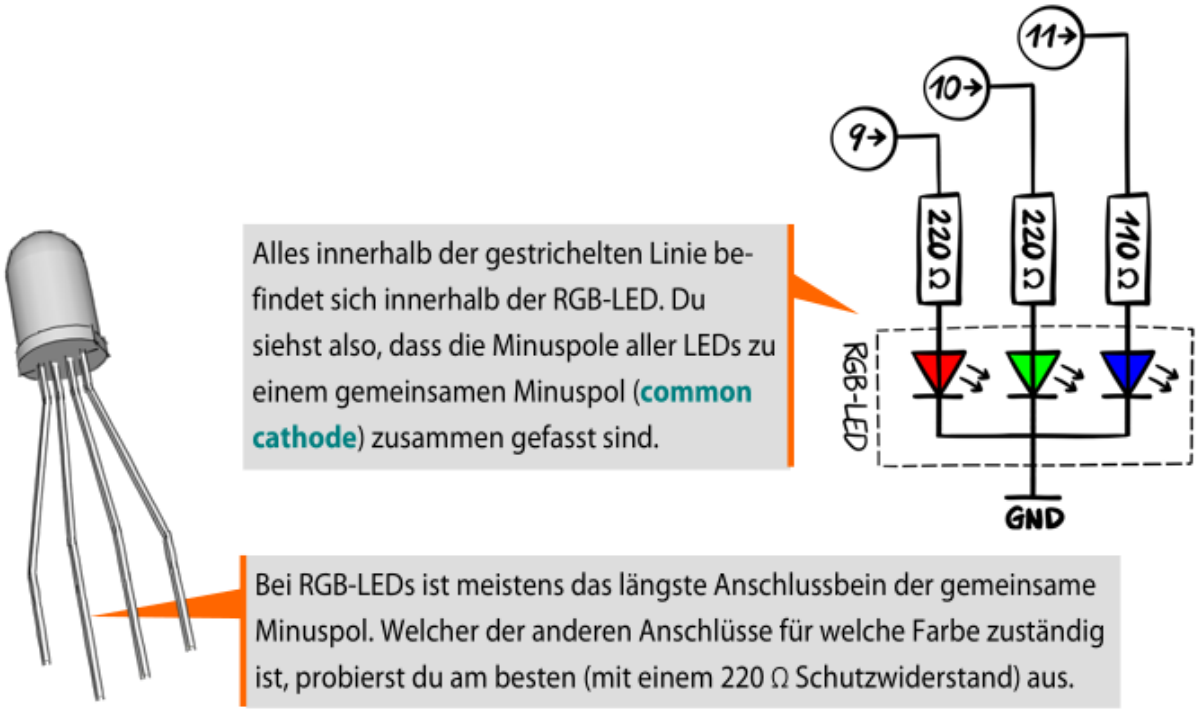
```
int s;

void setup() {
  randomSeed(analogRead(A1));
  Serial.begin(9600);
}

void loop {
  s=random(1,7);
  Serial.println(s);
  delay(1000);
}
```

Alles so schön bunt hier: RGB LEDs

Eine sogenannte RGB-LED beinhaltet drei Leuchtdioden in den Farben rot, grün und blau (daher RGB) in einem Gehäuse mit gemeinsamem Minuspol. Die Schaltung rechts zeigt dir, wie eine solche Leuchtdiode an geschlossen wird.



Warum rot, grün und blau? In unserem Auge werden die Farben mit Hilfe der Zapfen auf der Netzhaut wahrgenommen. Dabei gibt es drei Zapfentypen: Einen hauptsächlich für rot, einen für grün und einen für blau. Mit den drei Farben einer RGB-LED kann man die drei Zapfentypen gezielt anregen und so im menschlichen Auge jeden Farbeindruck hervorrufen. Das nutzen auch Handydisplays, Monitore und Fernseher. Dort besteht - wenn man ganz genau hinsieht - jeder Bildpunkt aus Lichtquellen in genau diesen drei Farben.

Aufgabe 9.5

Finde heraus, welcher der anderen Anschlüsse für welche Farbe ist und schließe die LED entsprechend dem obigen Schaltbild an. Welche Mischfarben ergeben sich, wenn rot und grün, rot und blau, grün und blau sowie alle drei LEDs an sind?

Aufgabe 9.6

Suche die etwa zehn Tippfehler in folgendem Programm.... Was soll das Programm machen?

```
void setup(); {
serialBegin(6900);
}

void loop {
for(int i=0;i<10;i=i+1){
 Serial.pint(“Huhu“);
 for(u=0,u<100;u=u+1){
```

```
    Serial.print(i);
    Serial.print(“,“);
    Serial.print(u);
  }
}
```

Aufgabe 9.7

Noch mehr Farbkombinationen als in Aufgabe 9.5 ergeben sich, wenn man die Farben mit `analogWrite()` gedimmt mischt. Schreibe ein Programm, das alle überhaupt möglichen Farbkombinationen anzeigt, also 256*256*256=16777216 Farben.

Dazu brauchst du wahrscheinlich drei for-Schleifen in einander.

ardulb1
[1)]
Normalerweise die Pins 3, 5,6,9, 10 und 11