

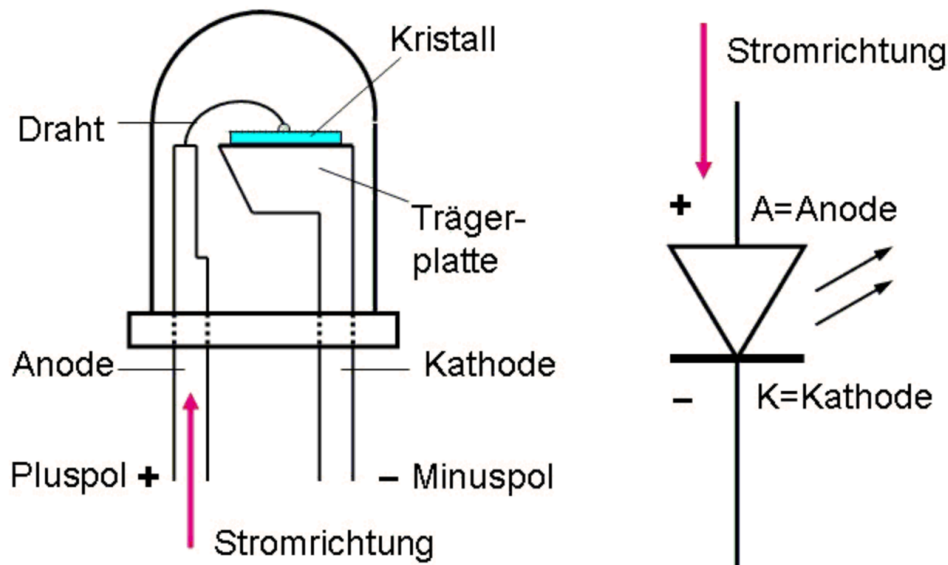
LED ansteuern

Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- LED

Beschreibung

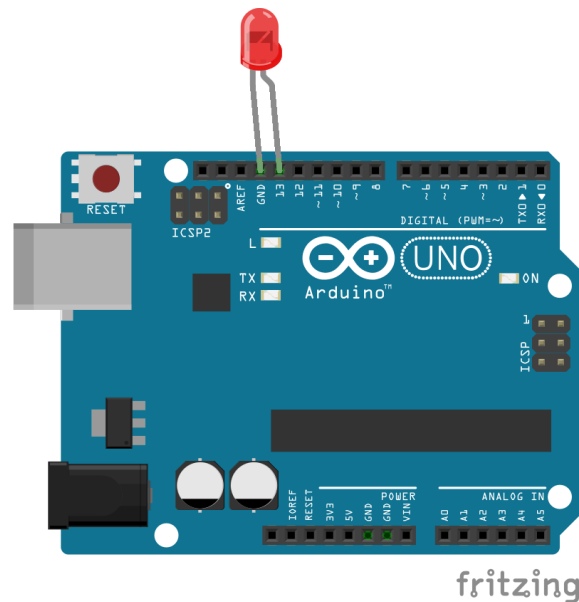
Die LED (light-emitting diode) ist ein elektrisches Bauteil, welches aufleuchtet, wenn ein elektrischer Strom in Durchlassrichtung fließt. Eine LED besitzt in der Regel zwei Beinchen. Das lange Bein ist die Anode und der Pluspol. Hier soll der Strom rein fließen, der von einer Stromquelle kommt. Das kurze Bein der LED ist die Kathode und der Minuspol. Hier soll der Strom raus und in den Ground fließen.



LED Aufbau

Verkabelung

Um die LED richtig mit dem Arduino Board zu verbinden, muss die Anode mit einem Pin und die Kathode mit einem Ground verbunden sein. Dies kann geschehen, indem man die Beine der LED direkt in die Pin-Anschlüsse des Arduinos setzt. Ebenso aber auch, indem man noch weitere Kabel oder ein Steckbrett zwischenschaltet. Die Hauptsache ist, dass die LED richtig herum an den Stromkreis angeschlossen wird.



Schaltplan LED

Code

```
// Der Pin Nr. 13 heißt von nun an LED
int LED = 13;

void setup() {
  // Der Pin LED soll ein Signal erhalten können (hier Strom). Er fungiert als
  // Akteur.
  pinMode(LED, OUTPUT);
}

void loop() {
  // Der Pin LED erhält das Signal, dass dort Strom fließen soll.
  digitalWrite(LED, HIGH);

  // Das Programm und das System verweilen in ihrem aktuellen Zustand für
  // 1000ms.
  delay(1000);

  // Der Pin LED erhält das Signal, dass dort kein Strom fließen soll.
  digitalWrite(LED, LOW);


  // Das Programm und das System verweilen in ihrem aktuellen Zustand für
  // 1000ms.
  delay(1000);
}
```

Der gezeigte Code lässt eine LED, die an den Pin 13 und den Ground angeschlossen ist, für 1 s = 1000 ms aufleuchten und für 1000 ms aus sein. Hierfür wird zunächst bestimmt, dass die LED an Pin 13 ist, und dieser von nun an im Code mit LED beschrieben werden kann.

Im void setup wird bestimmt, ob es sich bei der LED um einen Eingang oder einen Ausgang für ein Signal handelt. Da die LED ein Akteur ist, sie also auf ein Signal wartet und entsprechend des Signals agiert, muss sie als Ausgang bestimmt werden.

Im void loop wird dann der sich zu wiederholende Code geschrieben. Dieser soll in diesem Fall der LED für 1000 ms das ein Signal in Form eines Stromflusses geben und anschließend für 1000 ms dieses wieder Stromfluss unterbrechen. Der void loop wiederholt sich so lange wie eine Spannungsquelle an den Arduino angeschlossen ist.

Hochladen des Programms auf den Arduino

- Der Arduino wird mit dem USB Kabel am Rechner verbunden.
- **WICHTIG:** Wähle unter „Werkzeuge » PORT“ den COMX mit der höchsten Zahl aus (das ist der angeschlossene Arduino).
- Jetzt wird der Code auf den Arduino hochgeladen, indem man auf  drückt. Dabei ist darauf zu achten, dass nicht gleichzeitig die Batterie am Arduino angeschlossen ist.

Aufgaben

① Welches der Beine ist die Kathode und welches die Anode?

Bei der LED ist das lange Bein die Anode und das kurze Bein die Kathode.

② Woran wird die Kathode und woran die Anode angeschlossen?

Kathode ist der Minuspol und wird also an GND angeschlossen, die Anode ist der Pluspol und wird an einen Pin angeschlossen.

③ Baut den Code so um, dass die LED 5 Sekunden leuchtet und 2 Sekunden aus ist.

```
// Der Pin Nr. 13 heißt von nun an LED
int LED = 13;

void setup() {
  // Der Pin LED soll ein Signal erhalten können (hier Strom). Er fungiert als
  // Akteur.
  pinMode(LED, OUTPUT);
}

void loop() {
  // Der Pin LED erhält das Signal, dass dort Strom fließen soll.
  digitalWrite(LED, HIGH);

  // Das Programm und das System verweilen in ihrem aktuellen Zustand für
  // 5000ms.
  delay(5000);

  // Der Pin LED erhält das Signal, dass dort kein Strom fließen soll.
  digitalWrite(LED, LOW);

  // Das Programm und das System verweilen in ihrem aktuellen Zustand für
  // 2000ms.
  delay(2000);
}
```

Das Material und dessen Inhalte sind - sofern nicht anders angegeben - lizenziert unter der Creative Commons Lizenz CC BY-NC-SA 4.0 (für den vollständigen Lizenztext siehe <https://creativecommons.org/licenses/by-sa/4.0/legalcode>)