

# Einführung Arduino: 1. Eine LED pulsieren lassen

EinführungArduino\_1LED\_pulsieren | Arduino 1.8.9

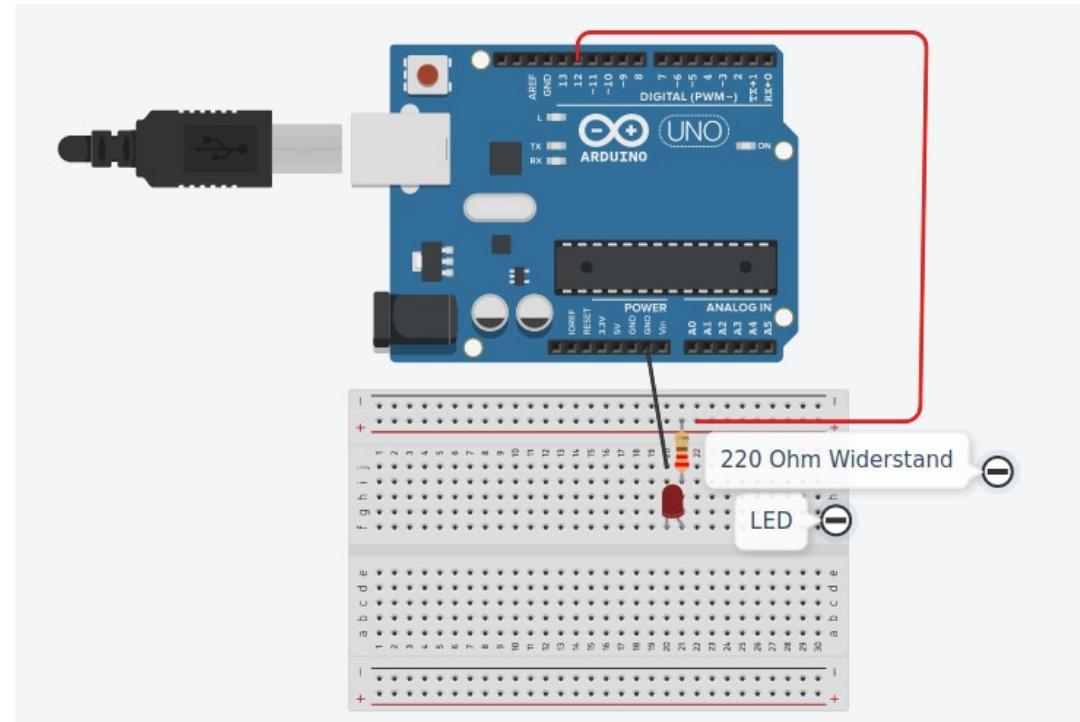
Datei Bearbeiten Sketch Werkzeuge Hilfe

EinführungArduino\_1LED\_pulsieren

```
1 int LedPin = 12; // Wir benennen den Pin 12 als LedPin
2
3
4
5 void setup()
6 {
7     pinMode(LedPin, OUTPUT); // Wir definieren den LedPin als Ausgang
8 }
9
10 void loop()
11 {
12
13     digitalWrite(LedPin, HIGH); // Wir schalten den Strom beim LedPin ein
14     delay(2000); // Der Strom bleibt für zwei Sekunden ein
15     digitalWrite(LedPin, LOW); // Wir schalten den Strom beim LedPin aus
16     delay(3000); // Der Strom bleibt für drei Sekunden aus
17 }
```

Speichern abgeschlossen.

8 Arduino/Genuino Uno auf /dev/ttyACM0



# Einführung Arduino: 2. Drei LEDs nacheinander

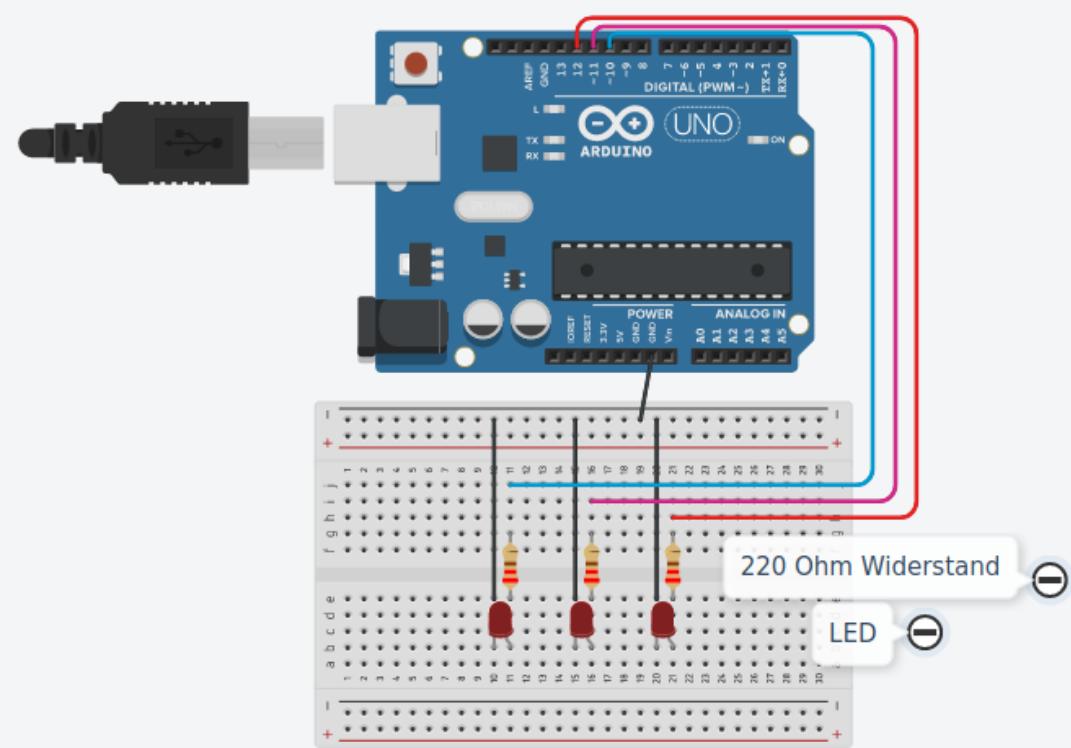
EinführungArduino\_3LED\_nacheinander | Arduino 1.8.9

Datei Bearbeiten Sketch Werkzeuge Hilfe

EinführungArduino\_3LED\_nacheinander

```
1 int LedPin = 12; // Wir benennen den Pin 12 als LedPin
2 int LedPin2 = 11; // Wir benennen den Pin 11 als LedPin2
3 int LedPin3 = 10; // Wir benennen den Pin 10 als LedPin3
4
5 void setup()
6{
7
8{
9 pinMode(LedPin, OUTPUT); // Wir definieren den LedPin als Eingang
10 pinMode(LedPin2, OUTPUT); // Wir definieren den LedPin als Eingang
11 pinMode(LedPin3, OUTPUT); // Wir definieren den LedPin als Eingang
12 }
13
14 void loop()
15
16{
17 digitalWrite(LedPin, HIGH); // Wir schalten den Strom beim LedPin ein
18 delay(1000); // Der Strom bleibt für zwei Sekunden ein
19 digitalWrite(LedPin, LOW); // Wir schalten den Strom beim LedPin aus
20 delay(500); // Der Strom bleibt für 500 Millisekunden aus
21
22 digitalWrite(LedPin2, HIGH); // Wir schalten den Strom beim LedPin2 ein
23 delay(1000); // Der Strom bleibt für zwei Sekunden ein
24 digitalWrite(LedPin2, LOW); // Wir schalten den Strom beim LedPin aus
25 delay(500); // Der Strom bleibt für 500 Millisekunden aus
26
27 digitalWrite(LedPin3, HIGH); // Wir schalten den Strom beim LedPin3 ein
28 delay(1000); // Der Strom bleibt für zwei Sekunden ein
29 digitalWrite(LedPin3, LOW); // Wir schalten den Strom beim LedPin aus
30 delay(500); // Der Strom bleibt für 500 Millisekunden aus
31 }
```

Arduino/Genuino Uno auf /dev/ttyACM0



# Einführung Arduino: 3. Piezo Lautsprecher Töne

EinführungArduino\_piezo\_tone | Arduino 1.8.9

Datei Bearbeiten Sketch Werkzeuge Hilfe

Einf\_HrungArduino\_piezo\_tone

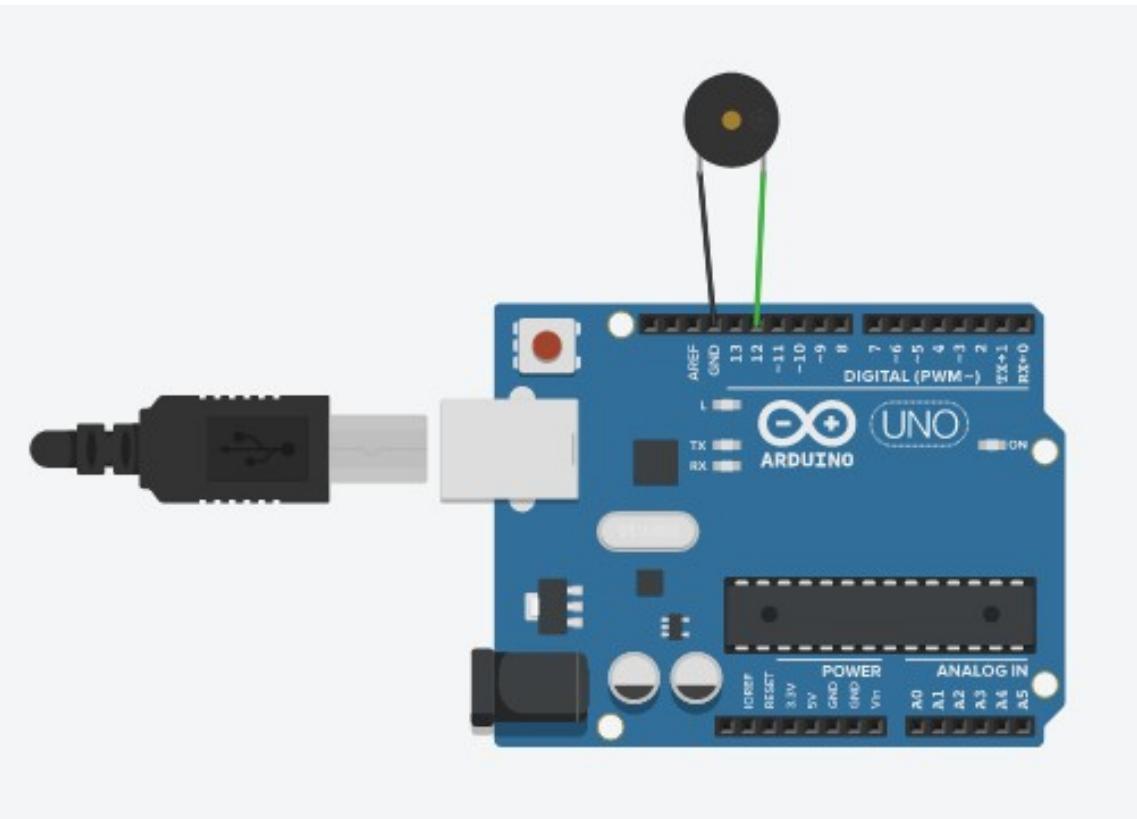
```
1 // Das "tone(X/Y)" Element Erzeugt einen Ton Y=Hz Frequenz am Pin X
2 // Folgend sind Variablen Definiert mit der entsprechenden Frequenz.
3 // Das heisst wenn man beim void loop beispielsweise NOTE_C4 eingibt
4 // an Stelle der direkten Frequenzzahl, wird dort die Note C4 gespielt.
5 // Diese und weiter Noten können aus dem Arduino Beispiel toneMelody - pitches.h
6 // aus dem Tab pitches h herauskopiert werden.
7 //
8 */
9 #define NOTE_C4 262
10 #define NOTE_C5 277
11 #define NOTE_D4 294
12 #define NOTE_D5 311
13 #define NOTE_E4 330
14 #define NOTE_F4 349
15 #define NOTE_F5 370
16 #define NOTE_G4 392
17 #define NOTE_G5 415
18 #define NOTE_A4 440
19 #define NOTE_A5 466
20 #define NOTE_B4 494
21
22 void setup() // Im Setup werden keine Informationen benötigt.
23 {}
24
25 void loop()
26{
27 tone(12, NOTE_C4); // Am Pin 12 wird ein Ton mit Frequenz von 262Hz erzeugt
28 // 262Hz ist die Frequenz der Note C4
29 delay(1000); // Der Ton hält eine Sekunde lang an
30 noTone(12); // Der Ton wird ausgeschalten
31 delay(1000); // Der Ton bleibt eine Sekunde aus
32 }
```

Speichern abgeschlossen.

Eckwerten müssen mit einem Buchstaben oder einer Zahl beginnen,  
gefolgt von Buchstaben, Zahlen, Bindestrichen, Punkten und Unterstrichen.  
Die maximale Länge beträgt 63 Zeichen.

1

Arduino/Genuino Uno auf /dev/ttyACM0



# Einführung Arduino: 4. LDR Sensor Fotowiderstand - Lichtwerte messen

EinführungArduino\_LDR | Arduino 1.8.9

Datei Bearbeiten Sketch Werkzeuge Hilfe

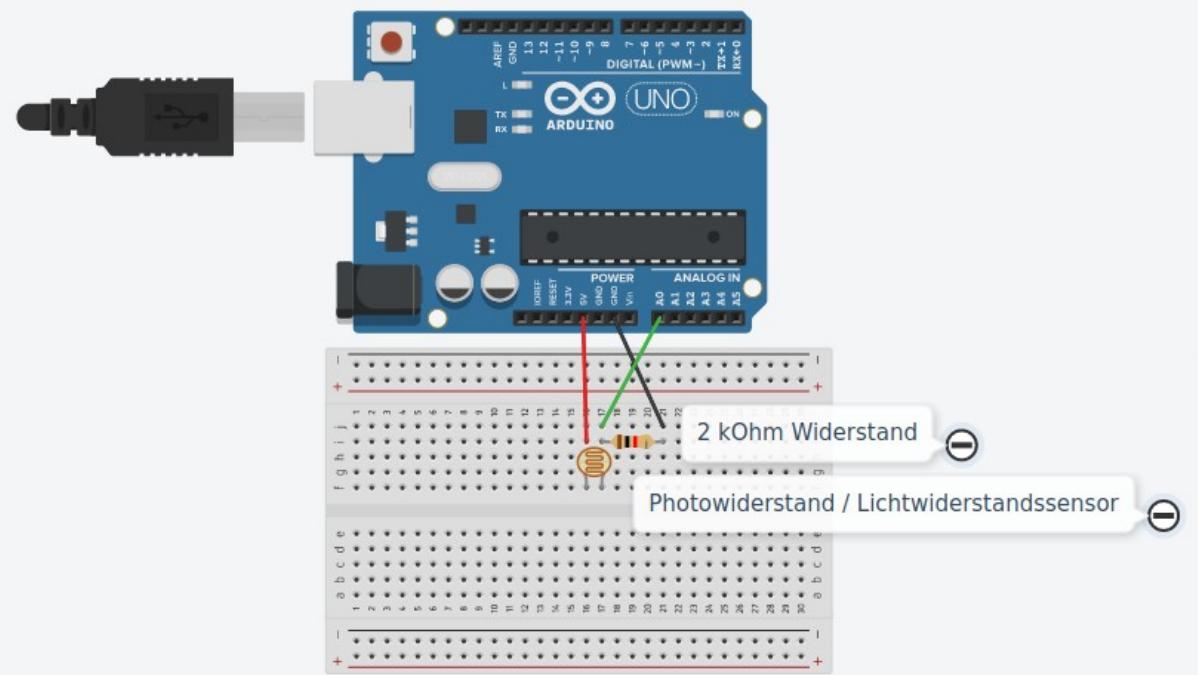
Einf\_hrungArduino\_LDR

```
1
2
3// LDR (Light Dependent Resistor), Fotowiderstand
4 * Je mehr Licht auf den Fotowiderstand fällt,
5 * desto kleiner wird sein elektrischer Widerstand
6 */
7
8
9 int LDRpin = A0; // Wir benennen den Pin 0 als LDRpin
10 int LDRWert = 0; // Wir erstellen eine Variable Namens LDRWert
11
12 void setup()
13{
14 Serial.begin(9600); // Die Verbindung zum seriellen Monitor wird integriert
15 }
16
17 void loop()
18 LDRWert = analogRead(LDRpin); // Der Wert vom LDRpin wird gelesen
19 Serial.println(LDRWert); // Der Wert wird an den seriellen Monitor gesendet
20 delay(100); // Wir warten 100 Millisekunden
21 }
```

avrdude done. Thank you.

5 Arduino/Genuino Uno auf /dev/ttyACM0

Hier öffnest du den virtuellen Monitor



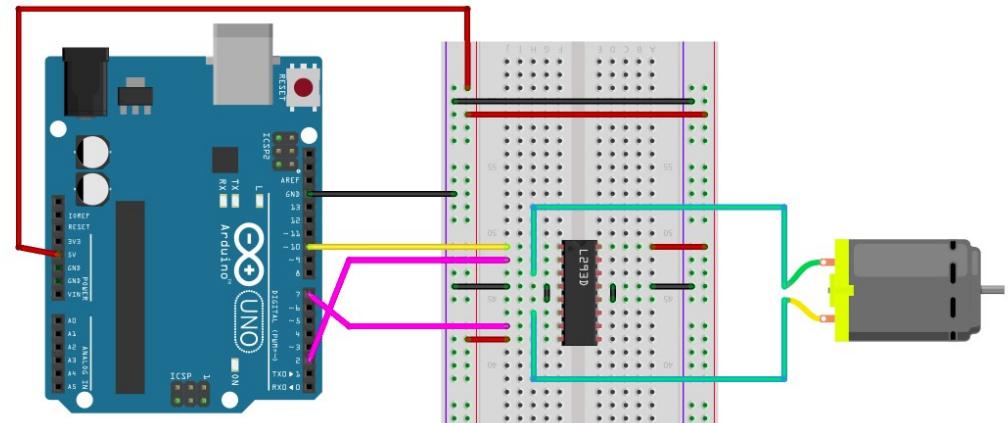
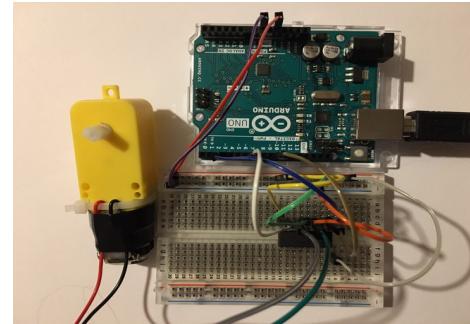
# Einführung Arduino: 5. Motor Motortreiber H-Brücke

EinführungArduino\_Motortreiber\_H-Brücke | Arduino 1.8.9

Datei Bearbeiten Sketch Werkzeuge Hilfe

```
1 // Wir verwenden für diese Schaltung eine H-Brücke L293D als Motortreiber
2
3 const int motorPin1 = 2; // Wir benennen den Pin 2 als motorPin1
4 const int motorPin2 = 7; // Wir benennen den Pin 7 als motorPin2
5 const int Aktivierungspin = 10; // Wir benennen den Pin 9 als Aktivierungspin
6 int speed = 125; // Geschwindigkeit 50%, max. 255
7
8 void setup() {
9     pinMode(motorPin1, OUTPUT); // Wir definieren den motorPin1 als Ausgang
10    pinMode(motorPin2, OUTPUT); // Wir definieren den motorPin2 als Ausgang
11 }
12 void loop() {
13     analogWrite(Aktivierungspin, speed);
14     // Aktivierungspin wird mit Geschwindigkeit (speed) aktiviert
15     digitalWrite(motorPin1, HIGH); digitalWrite(motorPin2, LOW);
16     // beim motorPin1 wird Strom an-, und beim motorPin2 abgeschaltet
17     delay(1000); // der Zustand bleibt für 1 Sekunde
18     digitalWrite(motorPin1, LOW); digitalWrite(motorPin2, HIGH);
19     // beim motorPin1 wird der Strom aus, und beim motorPin2 angeschaltet
20     delay(1000); // der Zustand bleibt für 1 Sekunde
21     analogWrite(Aktivierungspin, 0); // Motor aus
22     delay(4000); // der Zustand bleibt für 4 Sekunden
23 }
24 }
```

Speichern abgeschlossen.



# Einführung Arduino: if...else LED und Button

Einf\_hungArduino\_if\_else\_Button\_LED | Arduino 1.8.9

Datei Bearbeiten Sketch Werkzeuge Hilfe

Einf\_hungArduino\_if\_else\_Button\_LED

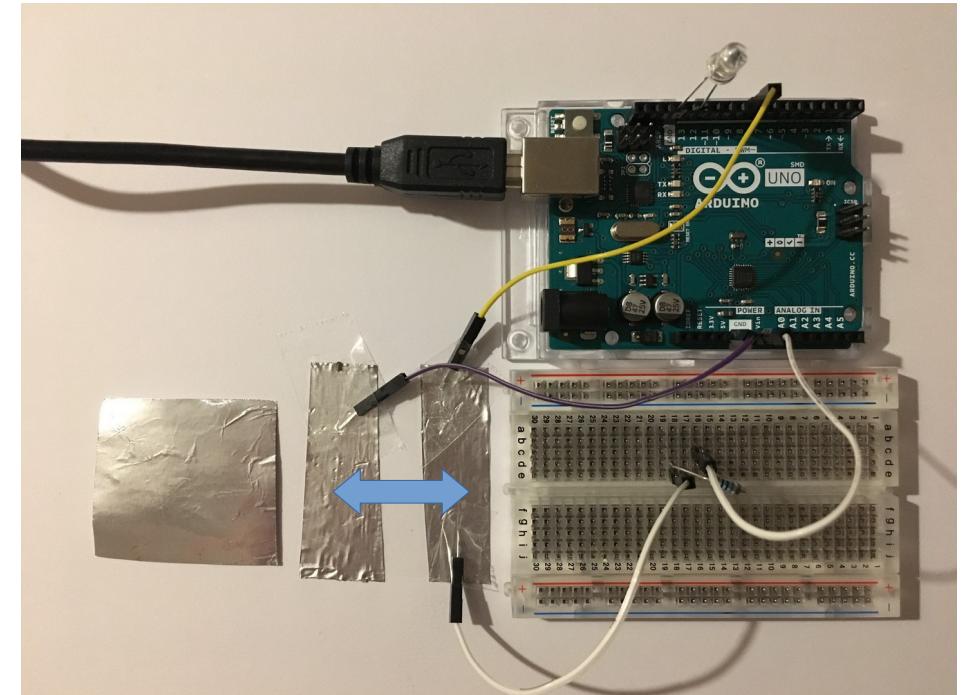
```
1 int buttonPin = 8; // Wir benennen den Pin 8 als buttonPin
2 int ledPin = 12; // Wir benennen den Pin 12 als ledPin
3
4
5 int buttonState = 0; // Variabel, um den Status des Button zu lesen
6
7 void setup()
8{
9  pinMode(ledPin, OUTPUT); // Wir definieren den ledPin als Ausgang
10 pinMode(buttonPin, INPUT); // Wir definieren den buttonPin als Eingang
11 }
12
13 void loop()
14
15{
16  buttonState = digitalRead(buttonPin); // buttonState liest den Status vom Button Pin
17
18  if (buttonState == HIGH) // Wenn der buttonState HIGH ist, also Strom am buttonPin fliessst..
19  {
20    digitalWrite(ledPin, HIGH); //... dann wird der ledPin angeschaltet
21    delay(500); // Der Zustand hält eine Sekunde an.
22  }
23
24  else // Wenn nicht...
25  {
26    digitalWrite(ledPin, LOW); // ... dann wird der ledPin ausgeschaltet
27    delay (500); // Der Zustand hält eine Sekunde an.
28  }
29 }
```

Speichern abgeschlossen.

avdude done. Thank you.

3

Arduino/Genuino Uno auf /dev/ttyACM0



## Anschlüsse Arduino:

- weisses Kabel → GND
- violettes Kabel → 5V
- gelbes Kabel → Pin8
- 1kOhm Widerstand

# Einführung Arduino: if ... else if LDR Sensor und LED

Einf\_hungArduino\_If\_Else\_LDR\_LED | Arduino 1.8.9

Datei Bearbeiten Sketch Werkzeuge Hilfe

Einf\_hungArduino\_If\_Else\_LDR\_LED

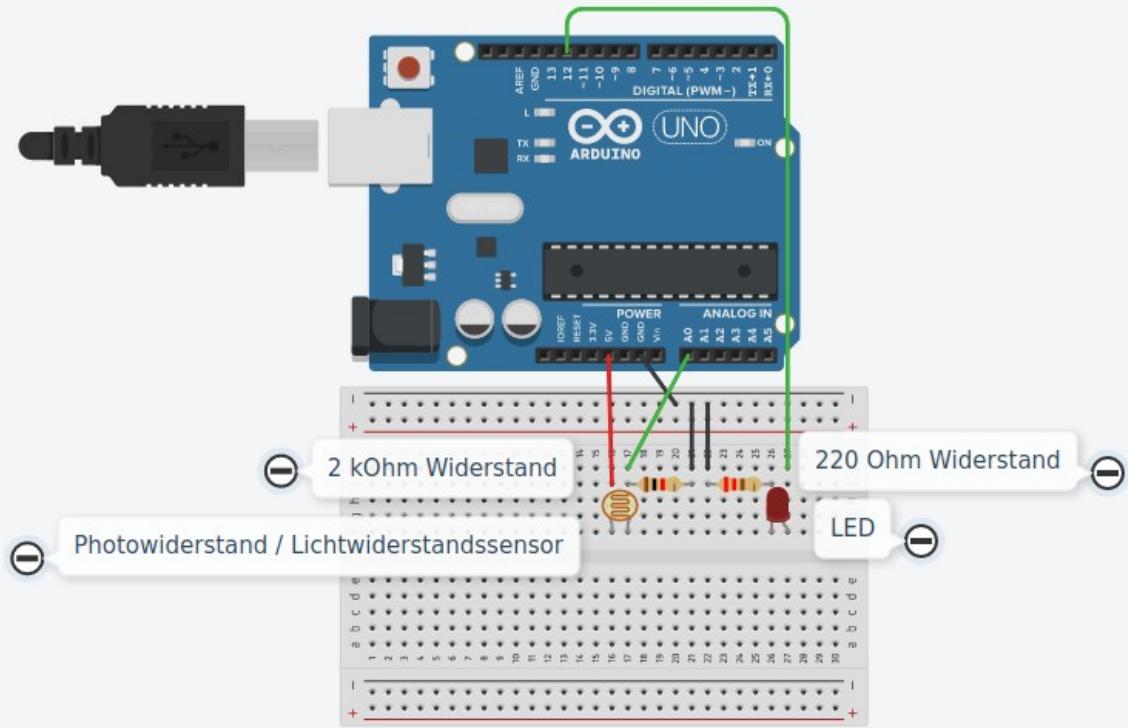
```
7
8
9 int LDRpin = A0; // Wir benennen den Pin 0 als LDRPin
10 int LDRWert = 0; // Wir erstellen eine Variable Namens LDRWert
11
12 int LedPin = 12; // Wir benennen den Pin 12 als LedPin
13
14 void setup()
15{
16  Serial.begin(9600); // Die Verbindung zum seriellen Monitor wird integriert
17  pinMode (LedPin, OUTPUT); // Wir definieren den LedPin ( Pin 12 ) als Ausgang
18 }
19
20 void loop()
21{
22  LDRWert = analogRead(LDRpin); // Der Wert vom LDRpin wird gelesen
23  Serial.println(LDRWert); // Der Wert wird an den Seriellen Monitor gesendet
24
25  if (LDRWert < 100) // Wenn der LDR Wert kleiner als 100 ist...
26  {
27    digitalWrite (LedPin, HIGH); // ... dann wird der LedPin angeschalten
28    delay(1000); // Der Zustand bleibt für eine Sekunde.
29  }
30
31  else if ( LDRWert > 100) // Wenn der LDR Wert grösser als 100 ist...
32  {
33    digitalWrite (LedPin, LOW); // ... dann wird der LedPin ausgeschalten
34    delay (1000); // Der Zustand bleibt für eine Sekunde.
35  }
36
37 }
```

Speichern abgeschlossen.

Der Sketch verwendet 2278 Bytes (7%) des Programmspeicherplatzes. Das Maximum sind 32256 Bytes. Globale Variablen verwenden 190 Bytes (9%) des dynamischen Speichers, 1858 Bytes für lokale Variablen.

20

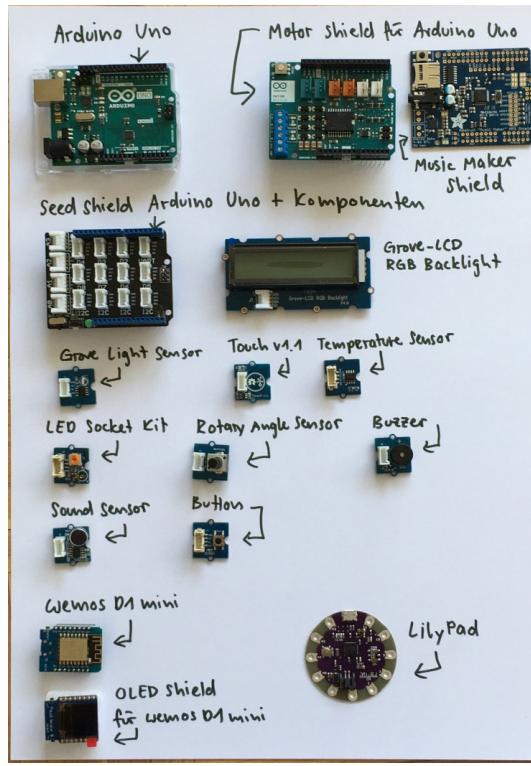
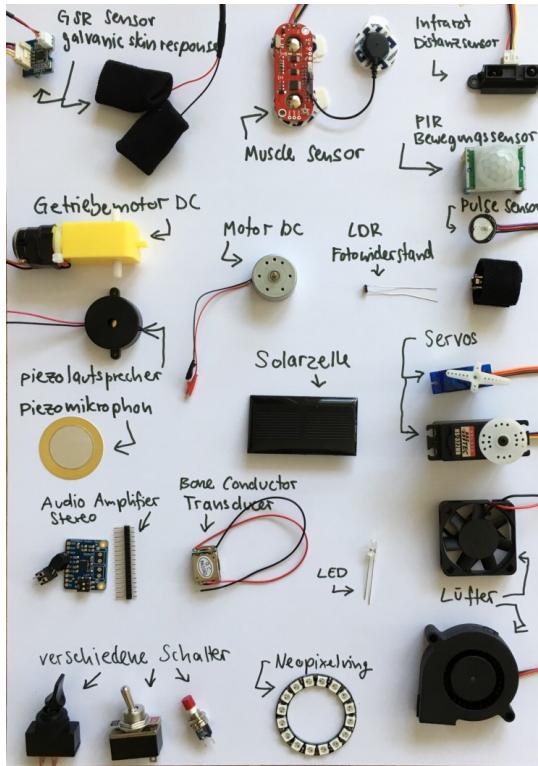
Arduino/Genuino Uno auf /dev/ttyACM0



# Einführung Arduino: Plattformen / Webseiten

- Hauptwebseite Arduino, Software, Projekte: [arduino.cc](http://arduino.cc)
  - Anleitungen von bestehenden Projekten auf Plattformen und Webseiten: [instructables.com](http://instructables.com), [kreativekiste.de](http://kreativekiste.de), [funduino.de](http://funduino.de) usw.
  - Viele Videotutorials auf [youtube.com](http://youtube.com)
- viele Arduino Projekte sind sehr gut dokumentiert, eigene Projekte entwickeln durch kopieren und anpassen von bestehenden zugänglichen Codes und Projekten
- viele Programme im ähnlichen Bereich bieten Schnittstellen zu Arduino  
Beispielsweise: Scratch, Processing, Pure Data usw.

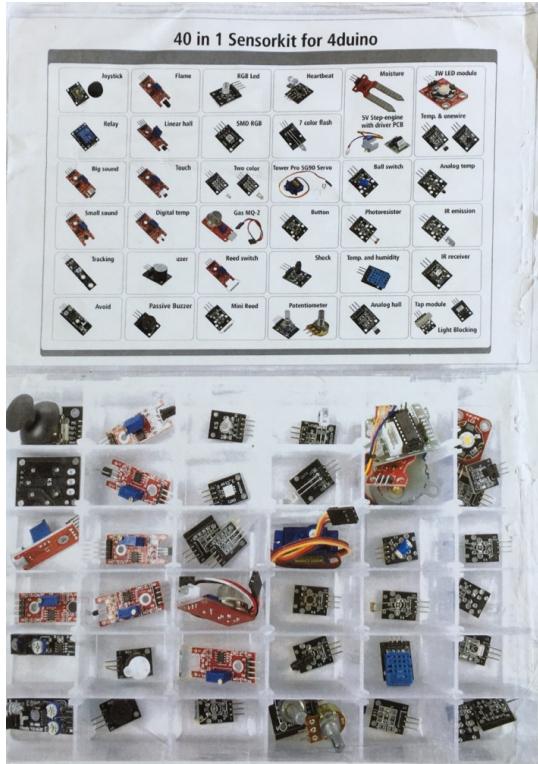
# Einführung Arduino: Komponenten, Kontroller und Shields



Es gibt verschiedene Komponenten, die man direkt über das Arduino ansteuern kann (OUTPUT) oder deren Daten man über das Arduino einlesen kann, so beispielsweise Sensordaten (INPUT).

Für verschiedene Mikrocontroller wie das Arduino Uno gibt es sogenannte Shields, die man direkt auf die Kontroller aufstecken kann und somit einen einfachen Zugang zu zusätzlichen Funktionen und Komponenten bekommt.

# Einführung Arduino: Komponenten, Kontroller und Shields



Hier seht ihr ein ganzes Sensorset, das wir für unser Modul von Digital Ideation zur Verfügung haben.

# Einführung Arduino: Wo kann ich Komponenten und Kontroller bestellen?

- Conrad.ch (Einzelteile auch vor Ort in der Filiale)
  - Bastelgarage.ch (sehr schnelle Lieferung, Leitung aus der Makerszene, Familienunternehmen, teilweise auch Anleitungen zu Komponenten, Support zu den Komponenten möglich)
  - Distrelec.ch ( auch günstige Komponenten, schnelle Lieferung)
  - AliExpress.com (sehr billige Komponenten direkt via China oder anderen Ländern, wo billig produziert wird, längere Lieferzeiten)
  - Pusterla Zürich (die Besitzer wissen sehr viel und können tollen Support bieten)
- Je nach Anforderungen empfehlen sich andere Läden und Online Shops