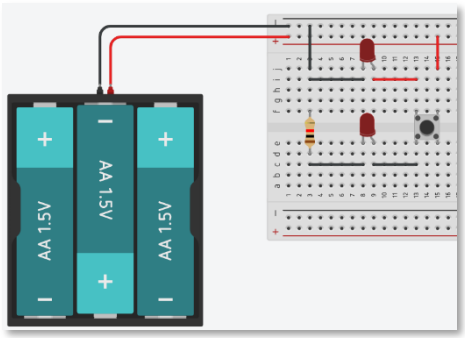
bs-lif.de Staatliche Berufsschule Lichtenfels	Thema: Mikrocontroller		
	Klasse:	Datum:	Fach: ITT

Einführung in Mikrocontroller

Mit Hilfe der Webseite tinkercad.com lassen sich elektronische Schaltungen und kleine Projekte mit dem Mikrocontroller-Board Arduino Uno R3 erstellen und im Rahmen einer Simulation ausführen. Auf diese Weise können Sie sich mit sehr geringem Aufwand mit den Grundlagen vertraut machen.

Lesen Sie zum besseren Verständnis diese Beschreibung zum grundsätzlichen Aufbau eines Arduino-Programmes: https://www.arduino.cc/en/Tutorial/BuiltInExamples/BareMinimum

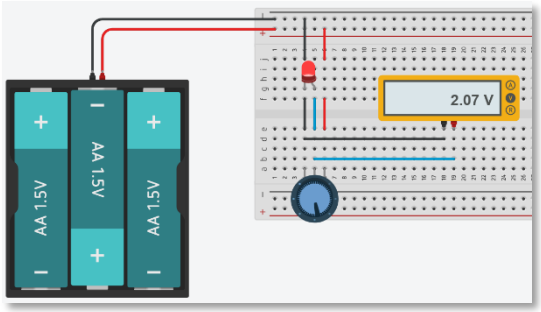
1. Elektrische Grundlagen



Erstellen Sie einen einfachen Schaltkreis mit einer 4,5V-Spannungsquelle, zwei parallel geschalteten LEDs und einem Taster. Schalten Sie einen 1 kΩ-Widerstand mit einer der LEDs in Reihe.

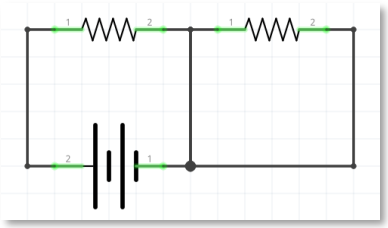
Wenn Sie den Taster drücken und der Stromkreis geschlossen wird, dann beginnen beide zu leuchten. TinkerCAD signalisiert ihnen bei einer der LEDs, dass 4,5V zu viel Spannung sind. Achten Sie deshalb bei realen Schaltkreisen stets auf Spannung und Stromfluss. Verwenden Sie z. B. wie hier gezeigt einen Vorwiderstand.

2. Spannungsteilung durch Widerstände



Bei zwei in Reihe geschalteten Widerständen teilt sich die Spannung im Verhältnis der beiden Widerstände. Durch die Verwendung eines Potentiometers, lässt sich dieses Verhältnis und damit die anliegende Spannung regeln.

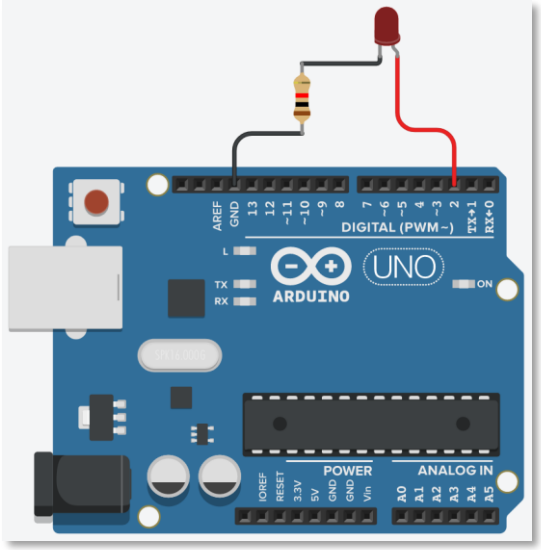
Ein Potentiometer kann man sich als eine Verkettung zweier Widerstände vorstellen, deren Gesamtwiderstand zwar gleich bleibt, deren Verhältnis zueinander sich jedoch durch Drehen am Regler verändern lässt.



Erstellen Sie eine Schaltung mit einem Potentiometer, an dessen eine Seite eine LED und ein Spannungsmessgerät angeschlossen sind und beobachten Sie die Veränderung der Spannung beim Verstellen des Potentiometers.

bs-lif.de Staatliche Berufsschule Lichtenfels	Thema: Mikrocontroller		
	Klasse:	Datum:	Fach: ITT

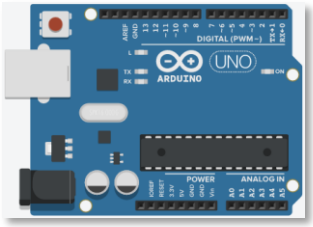
3. Der Arduino-Mikrocontroller



Erstellen Sie einen Schaltkreis mit einem Arduino-Board und einem LED mit Vorwiderstand (1 kΩ). Lassen Sie das angeschlossene und das eingebaute LED blinken.

Sie können sich bei Bedarf Hilfestellung bei dem Code-Beispiel im Anhang holen. Nehmen Sie sich die Zeit zu verstehen, was die einzelnen Code-Zeilen bewirken und kopieren Sie den Code nicht nur.

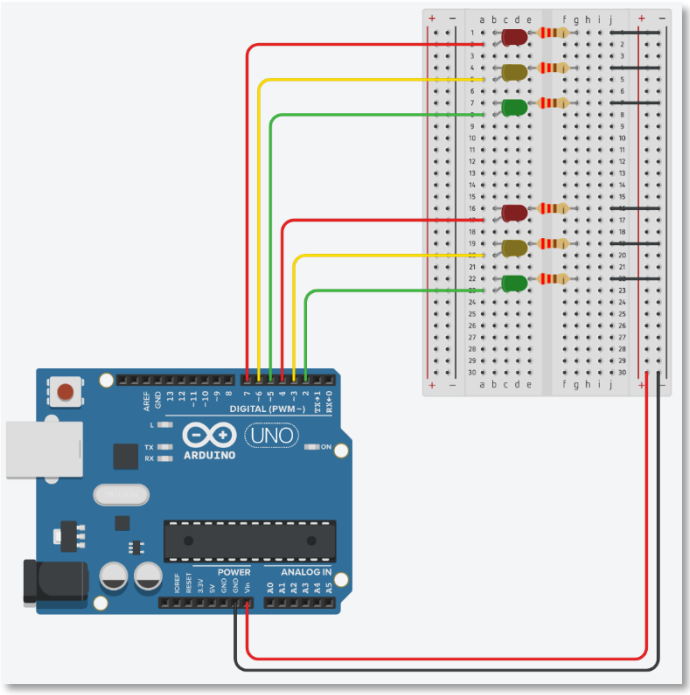
4. Serieller Monitor



Wenn ihr Mikrocontroller-Board per USB an Ihren Rechner angeschlossen ist, können Sie diese Verbindung nutzen um Konsolenausgaben zu machen. Dabei ist darauf zu achten, dass die gewählte Baudrate auf beiden Seiten übereinstimmt.

Nutzen Sie das Beispiel im Anhang um sich damit vertraut zu machen.

5. Ampelsteuerung

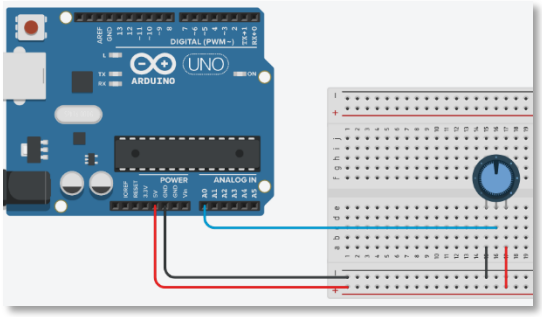


Mit dieser Aufgabe sollen Sie ein kleines Projekt umsetzen, nämlich die Steuerung einer Ampelanlage. Erstellen Sie ein Breadboard-Layout mit zwei Ampeln, die jeweils aus 3 LEDs bestehen. Als Wert für den Vorwiderstand eignen sich 220 Ω.

Vollständiger Ampelzyklus

Dauer	Ampel 1	Ampel 2
0,5 s	rot	rot-gelb
3,5 s	rot	grün
1 s	rot	gelb
0,5 s	rot-gelb	rot
3,5 s	grün	rot
1 s	gelb	rot

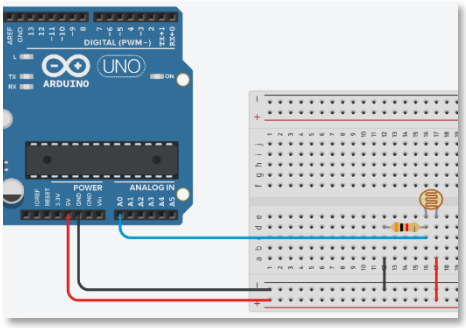
6. Einlesen von Messwerten: Potentiometer



In den bisherigen Beispielen haben wir nur Ausgaben gemacht. Nun sollen auch Messwerte eingelesen werden. Bauen Sie eine Schaltung mit einem Potentiometer auf und geben Sie den gemessenen Wert auf der seriellen Konsole aus.

Beispiel-Code hierzu finden Sie wieder im Anhang. Legen Sie den Schwerpunkt wie immer auf das Verständnis des Codes.

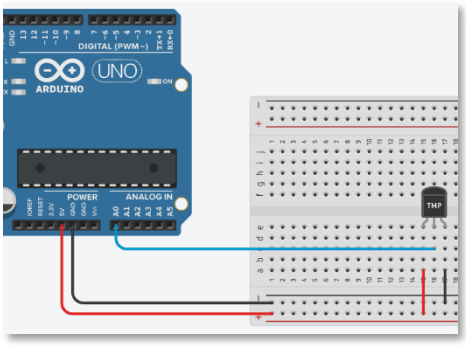
7. Einlesen von Messwerten: Photowiderstand



Auf die gleiche Weise wie bei Beispiel 6 können Sie auch mit Hilfe eines lichtempfindlichen Widerstandes (Photowiderstand, LDR) die Lichteinstrahlung messen. Für den festen Widerstand eignet sich ein Wert von ca. 3 kΩ.

Zeigen Sie den Messwert wieder auf der seriellen Konsole an.

8. Einlesen von Messwerten: Temperatursensor

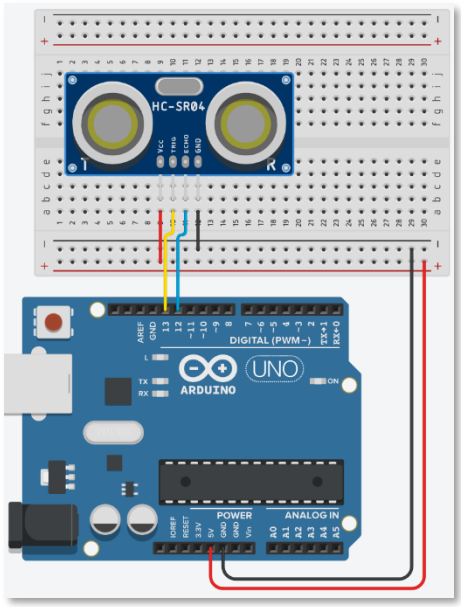


Wenn Sie in den beiden vorangegangenen Schaltungen das Potentiometer oder Widerstand und LDR durch einen Temperatursensor ersetzen, können Sie auf der seriellen Konsole beobachten, wie sich der Messwert ändert.

Welche Schwierigkeit können Sie bei dieser Art der Temperaturmessung beobachten?

Wie könnte man diese lösen?

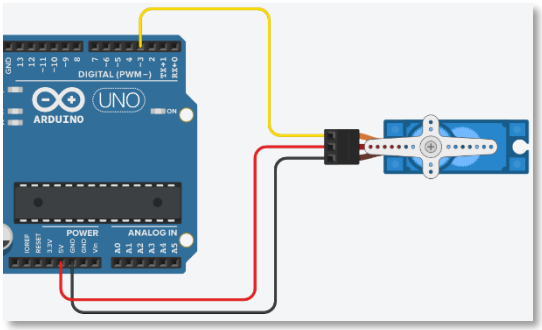
9. Entfernungsmessung mittels Ultraschalls



Ein sehr günstiger und weit verbreiteter Sensor zur Messung von kurzen Entfernungen ist das Modul HC-SR04. Es beruht auf dem Prinzip, dass das Modul über einen kurzen Impuls am TRIG-Pin dazu veranlasst wird, eine Reihe von Schallsignalen auszusenden. Das Eintreffen der reflektierten Signale wird über den ECHO-Pin signalisiert. Über die Schallgeschwindigkeit lässt sich aus dieser Laufzeit die Entfernung zum gemessenen Objekt berechnen.

Bilden Sie diesen Schaltkreis nach und ergänzen Sie das beiliegende Codebeispiel so, dass die gemessene Entfernung in der Konsole ausgegeben wird.

10. Steuerung eines Nicht-binären Aktors



Die Aktorsteuerung in den bisherigen Beispielen beschränkte sich auf das An- oder Ausschalten eines Ausganges, also auf binäre Aktionen.

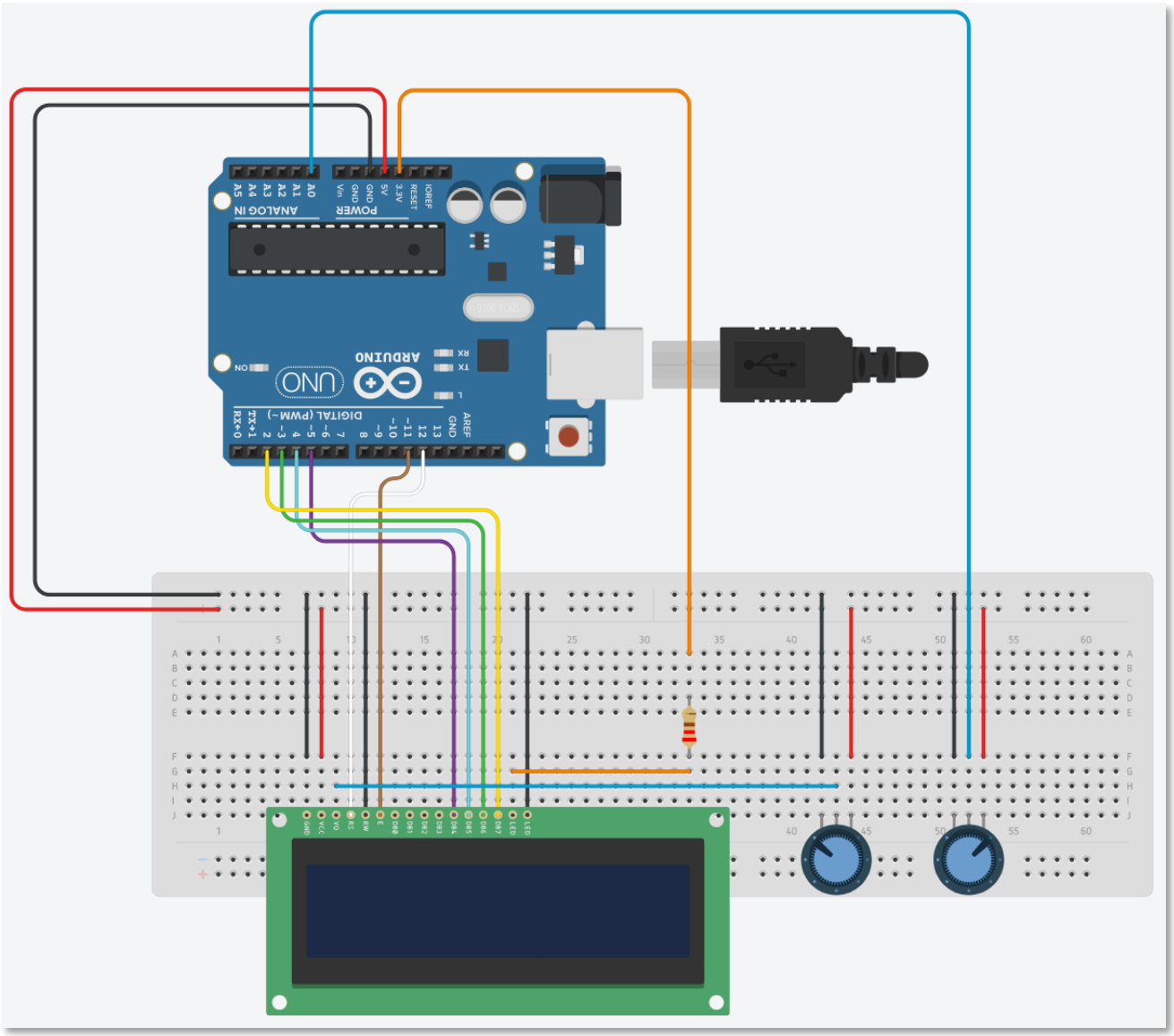
In diesem Beispiel steuern Sie ein Servo mittels Pulsweitenmodulation[1] an. Orientieren Sie sich am entsprechenden Codebeispiel im Anhang und erweitern Sie dieses so, dass sich das Servo in einer Endlosschleife stetig in kleien Schritten hin- und herdreht.

Geben Sie die jeweilige Servostellung als Zahlenwert auf der seriellen Konsole aus.

[1] https://www.elektronik-kompendium.de/sites/kom/0401111.htm

11. Visuelle Ausgabe: LCD-Display

Damit der Mikrocontroller nicht auf eine USB-Verbindung zu einem PC angewiesen ist um Informationen auszugeben, kann ein Display angesteuert werden. In diesem Beispiel wird ein 16x2-LCD-Display verwendet (16 Zeichen, 2 Zeilen). Dieses wird parallel angeschlossen. Es gibt auch die Möglichkeit ein Display über eine serielle Schnittstelle (I2C) anzuschließen, damit weniger Anschlusskanäle benötigt werden.



Anschlussbelegung

Display	**Arduino**
RS	12
RW	GND
E	11
LED-Kathode	GND
LED-Anode	3,3 V (220 Ω)

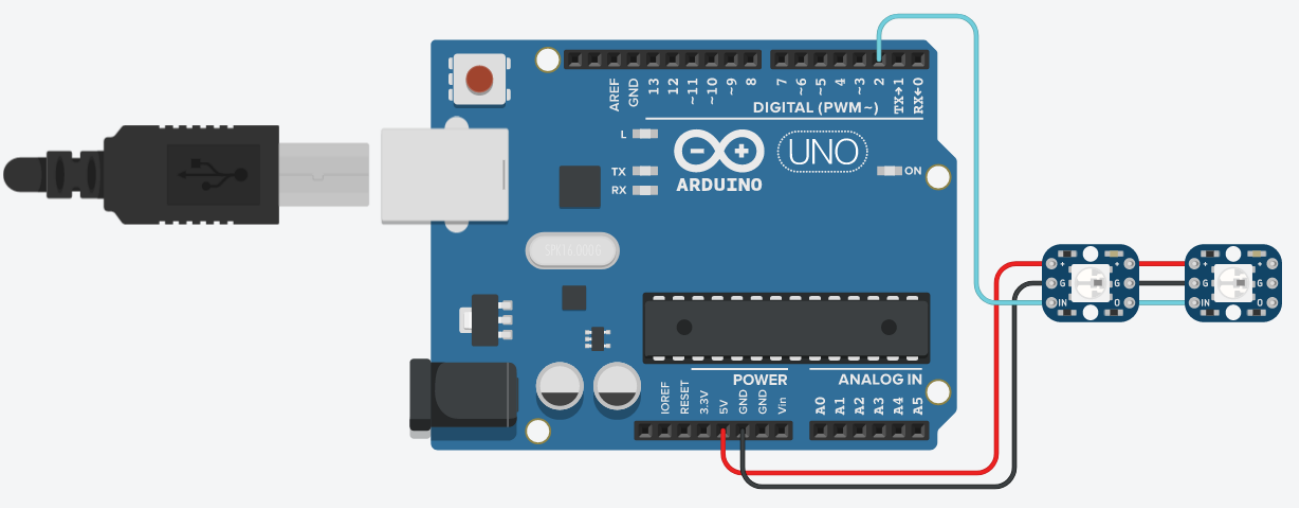
Display	**Arduino**
DB4	5
DB6	4
DB7	3
DB6	2
V0 (Kontrast)	variabel (470 Ω)

Erstellen Sie mit Hilfe des Codebeispiels im Anhang eine Schaltung, die einen Messwert von einem Potentiometer einliest und diesen auf dem LCD-Display ausgibt.

12. NeoPixel RGB LEDs

Leds anzusteuern kann von der Verkabelung sehr aufwändig sein, wie du in Aufgabe 5 mit der Ampelsteuerung sehen konntest. RGB-LEDs anzusteuern, bei der man nicht eine, sondern 3 Zuführungen braucht wegen der 3 Farben, verkomplizieren die Situation noch. Einiges leichter machen es einem da die NeoPixel-LEDs. Diese benötigen nur eine Stromversorgung und eine einzige Leitung für die Daten. Jedes Pixel hat einen Datenein- und einen -ausgang. So können viele LED hintereinandergeschaltet werden. Diese werden auch bereits vormontiert auf Streifen hergestellt.

In diesem Beispiel sind nur zwei einzelne NeoPixel-LEDs miteinander verbunden. Der Beispiel-Code wechselt im 2 Sekundentakt die Farben bei beiden LEDs. Wenn du möchtest, kannst du versuchen mit Hilfe von Drehpotentiometern an den analogen Eingängen die Farbe frei wählbar zu machen.



Anhang

Codebeispiel zu 3.

```
#define LED_EXTERN 2

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(LED_EXTERN, OUTPUT);

  digitalWrite(LED_EXTERN, LOW);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  digitalWrite(LED_EXTERN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  digitalWrite(LED_EXTERN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Codebeispiel zu 4.

```
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);

  // Serielle Ausgabe mit 115200 Baud starten
  Serial.begin(115200);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  Serial.println("LED ist an");
  delay(1000); // Wait for 1000 millisecond(s)


  digitalWrite(LED_BUILTIN, LOW);
  Serial.println("LED ist aus");
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Codebeispiel zu 6.

```
#define POTIPIN A0

void setup()
{
  Serial.begin(115200);
}

void loop()
{
  Serial.println(analogRead(POTIPIN));
  delay(100);
}
```

Codebeispiel zu 9.

```
#define PIN_TRIGGER 13
#define PIN_ECHO 12
#define SENSOR_MAX_RANGE 300

unsigned long time_of_flight;
unsigned int distance;

void setup()
{
  Serial.begin(115200);
  pinMode(PIN_TRIGGER, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
}

void loop()
{
  // 345m/s, Hin- und Rückweg, Messung in Mikrosekunden, 0,0345 cm/us
  float factor = 0.0345 / 2;
  unsigned int timeout = (SENSOR_MAX_RANGE / factor) * 1.1;

  digitalWrite(PIN_TRIGGER, LOW);
  delayMicroseconds(3);

  digitalWrite(PIN_TRIGGER, HIGH);
  delayMicroseconds(10);
  digitalWrite(PIN_TRIGGER, LOW);

  time_of_flight = pulseIn(PIN_ECHO, HIGH, timeout);
  distance = time_of_flight * factor;

  // Ausgeben der Entfernung, sofern diese in einem sinnvollen Bereich
  // (positiv und unter der Messgrenze des Sensors) liegt, oder eine
  // Fehlermeldung ausgeben

  delay(1000);
}
```

Codebeispiel zu 10.

```
#define SERVO 3
#define SERVO_UP 1
#define SERVO_DOWN 2

static int servoPos = 0;
static int servoDir = SERVO_UP;


void setup()
{
  pinMode(SERVO, OUTPUT);
  Serial.begin(115200);

  // Probelauf
  analogWrite(SERVO, 0);
  delay(500);
  analogWrite(SERVO, 127);
  delay(500);
  analogWrite(SERVO, 255);

  Serial.println("Setup abgeschlossen.");
}

void loop()
{
  // drehen und ausgeben
}
```

Codebeispiel zu 11.

```
#include <LiquidCrystal.h>

// Erstellen eines LCD unter Angabe der verwendeten Pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
  lcd.begin(16,2);
  pinMode(A0, INPUT);
}

void loop()
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("BS Lichtenfels");
  lcd.setCursor(0, 1);
  lcd.print("Messwert: ");
  delay(100);
}
```

Codebeispiel zu 12.

```
#include <Adafruit_NeoPixel.h>

#define NEO_PIXEL 2  // Pin für NeoPixel-Daten
#define NPIXELS 2  // Anzahl der NeoPixel

Adafruit_NeoPixel px = Adafruit_NeoPixel(NPIXELS, NEO_PIXEL, NEO_GRB +
NEO_KHZ800);

void setup()
{
  px.begin();
  px.clear();
  px.setBrightness(255);
  px.show();
  Serial.begin(9600);
  Serial.println("Setup");
}

void loop()
{
  Serial.println("One");
  px.setPixelColor(0, px.Color(255, 0, 0));
  px.setPixelColor(1, px.Color(255, 255, 0));
  px.show();
  delay(2000);

  Serial.println("Two");
  px.setPixelColor(0, px.Color(0, 255, 0));
  px.setPixelColor(1, px.Color(0, 255, 255));
  px.show();
  delay(2000);

  Serial.println("Three");
  px.setPixelColor(0, px.Color(0, 0, 255));
  px.setPixelColor(1, px.Color(255, 0, 255));
  px.show();
  delay(2000);
}
```