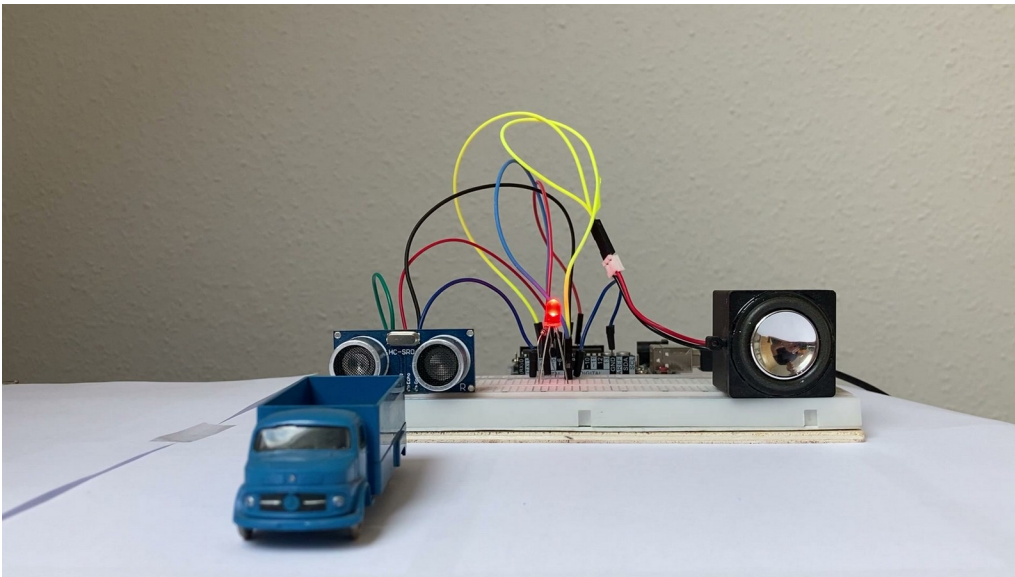
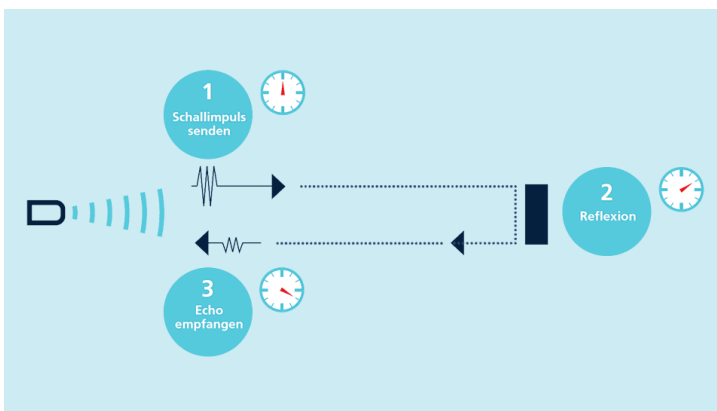


Bewegt sich ein Objekt auf den Entfernungsmesser zu, soll je nach dem noch verfügbaren Abstand zum Entfernungsmesser ein akustisches und optisches Signal gegeben werden.



Der Ultraschallsensor arbeitet nach einem einfachen Prinzip:



In regelmäßigen Abständen wird eine Schallwelle gesendet. Trifft sie auf einen Gegenstand wird die Schallwelle reflektiert und gelangt als Echo zurück. Mithilfe der Zeitspanne zwischen dem Aussenden des Signals und dem Wiedereintreffen lässt sich die Entfernung berechnen.

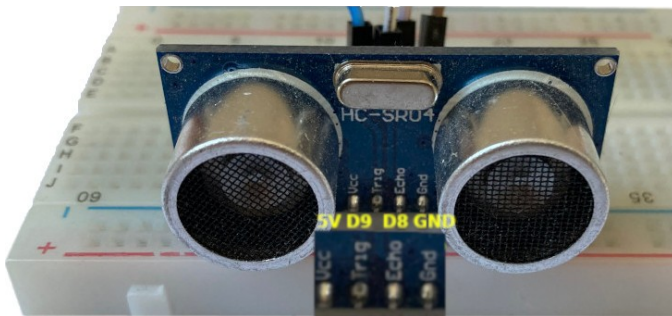
<https://www.microsonic.de/de/service/ultraschallsensoren/prinzip.htm>

Im Programm sendet der Befehl `pulseIn()` ein HIGH-Signal, startet einen Timer und wartet anschließend auf das zurückkommende Signal (bis es den Wert LOW hat). Daraufhin wird der Timer gestoppt und die Zeitspanne zwischen dem Senden des Signal (Trigger) und seiner Rückkehr (Echo) wird in Mikrosekunden ermittelt.

Umrechnung in cm:	$343,2 \text{ m} \cdot 100 = 34.320 \text{ cm}$
Strecke pro ms (Millisekunde):	$343.000 : 1.000 = 34,32 \text{ cm/ms}$
Strecke pro μs (Mikrosekunde):	$34,3 : 1.000 = 0,03432 \text{ cm}/\mu\text{s}$

Benötigte Bauteile:

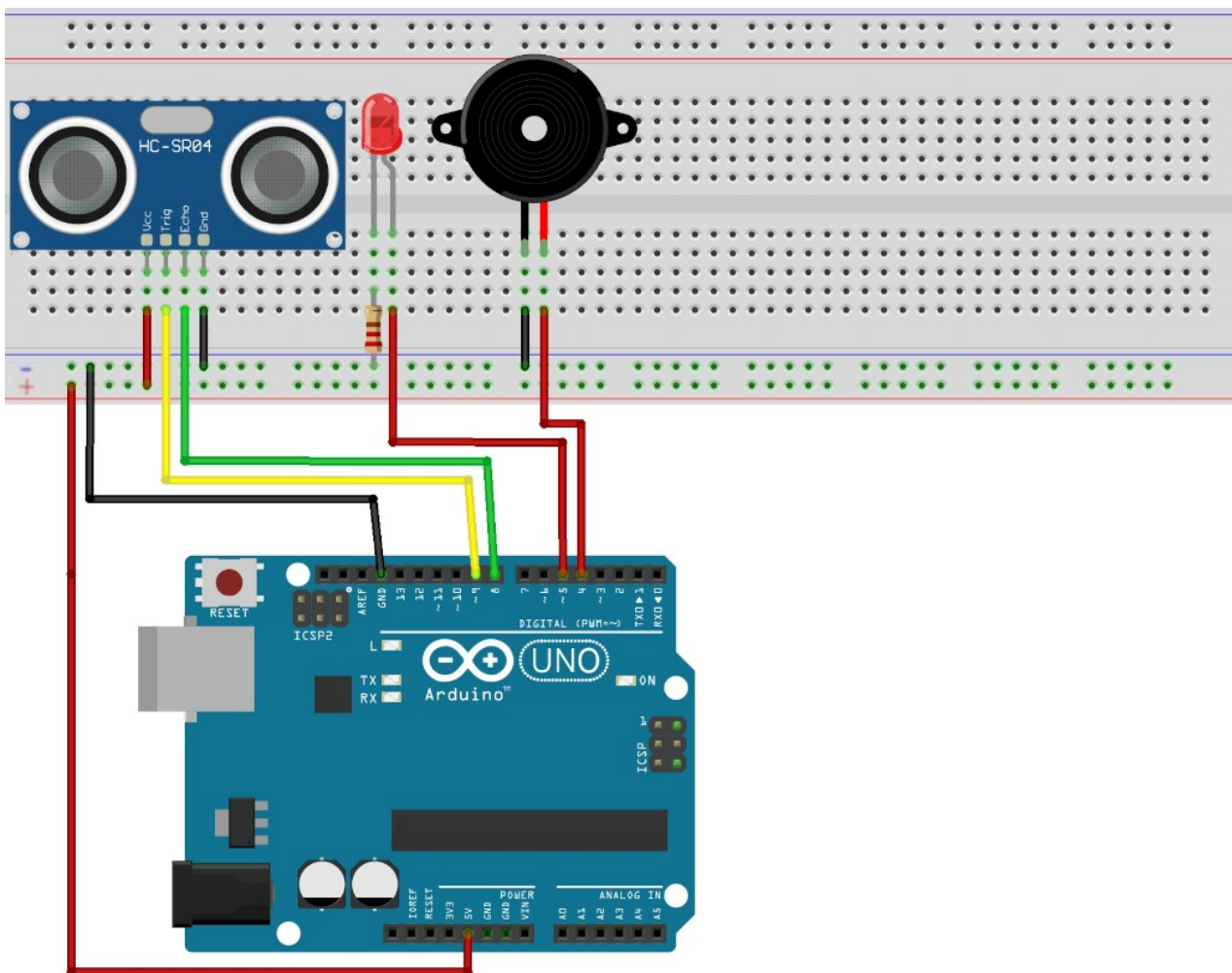
- rote LED
- Lautsprecher
- Widerstand 220 Ω
- Ultraschallsensor HC-SR04
- Leitungsdrähte



Trg = senden Echo = empfangen

Damit der Ultraschallsensor ordnungsgemäß arbeiten kann, musst du ihn so einstecken, dass Sender und Empfänger nach vorn zeigen, damit das Signal ungehindert gesendet und empfangen werden kann. Die Leitungsdrähte werden auf der Rückseite eingesteckt.

Baue die Schaltung auf.



Der Ultraschallsensor misst die Entfernung zum Hindernis.
Je kleiner der Abstand wird, desto schneller blinkt die LED und die Abstände zwischen den Tonsignalen werden kleiner.

Abstand zum Entfernungsmesser in cm	Abstand zwischen den Signalen in Millisekunden
13 bis 15	700
11 bis 12	500
9 bis 10	300
7 bis 8	200
5 bis 6	100
2 bis 4	0

Im Kopf werden die Variablen definiert. Außerdem soll die Zuweisung der Variablen mit define erfolgen.

```
# define LAUTSPRECHER 4
# define LED 5
# define SENDEN 9
# define ECHO 8
```

Der setup-Teil definiert die Bauteile als Eingabe-/Ausgabegeräte.

```
void setup()
{
  pinMode(SENDEN, OUTPUT);
  pinMode(ECHO, INPUT);
  pinMode(LAUTSPRECHER, OUTPUT);
  pinMode(LED, OUTPUT);
}
```

Die Funktion EntfernungMessen() misst die Entfernung des Objekts zum Entfernungsmesser.

```
int EntfernungMessen()
{
  long Entfernung = 0;

  // Sender kurz ausschalten um Störungen des Signal zu vermeiden
  digitalWrite(SENDEN, LOW);
  delay(5);

  // Signal senden
  digitalWrite(SENDEN, HIGH);
  delayMicroseconds(10);
  digitalWrite(SENDEN, LOW);
```

```
// pulseIn -> Zeit messen, bis das Signal zurückkommt
long Zeit = pulseIn(ECHO, HIGH);

// Entfernung in cm berechnen
Entfernung = (Zeit / 2) * 0.03432;
return Entfernung;
}
```

Der loop-Teil sorgt dafür, dass je nach gemessener Zeit der Lautsprecher im Intervall piepst und die LED im gleichen Intervall kurz aufleuchtet. switch fragt einen Wertebereich ab.

Beachte, dass vor und nach den drei Punkten ein Leerzeichen stehen muss.

```
void loop()
{
  // Funktion aufrufen
  long Entfernung = EntfernungMessen();

  // versuchen Messfehler auszuschließen
  if (Entfernung < 100)
  {
    switch (Entfernung)
    {
      case 13 ... 15:
        tone(LAUTSPRECHER, 1000, 2);
        digitalWrite(LED, HIGH);
        delay(50);
        digitalWrite(LED, LOW);
        delay(500);
        break;

      case 11 ... 12:
        tone(LAUTSPRECHER, 1000, 2);
        digitalWrite(LED, HIGH);
        delay(50);
        digitalWrite(LED, LOW);
        delay(400);
        break;

      case 9 ... 10:
        tone(LAUTSPRECHER, 1000, 2);
        digitalWrite(LED, HIGH);
        delay(50);
        digitalWrite(LED, LOW);
        delay(300);
        break;
    }
  }
}
```

```
case 7 ... 8:
  tone(LAUTSPRECHER, 1000, 2);
  digitalWrite(LED, HIGH);
  delay(50);
  digitalWrite(LED, LOW);
  delay(200);
  break;

case 5 ... 6:
  tone(LAUTSPRECHER, 1000, 2);
  digitalWrite(LED, HIGH);
  delay(50);
  digitalWrite(LED, LOW);
  delay(100);
  break;

case 2 ... 4:
  tone(LAUTSPRECHER, 1000);
  digitalWrite(LED, HIGH);
  break;
}
}
}
```