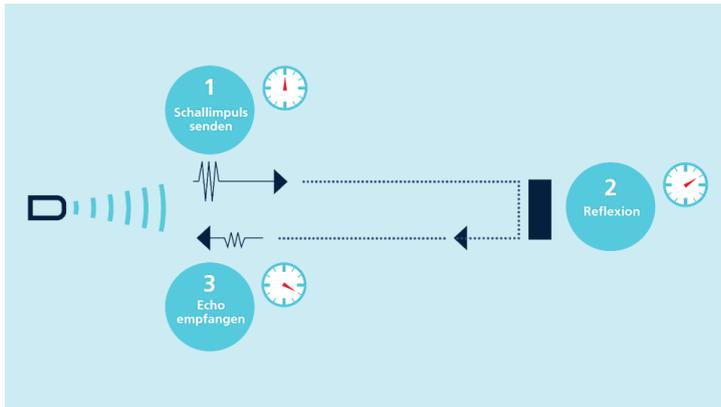


Der Ultraschallsensor soll als "Synthesizer" dienen: je nach Entfernung soll ein Ton gespielt werden.



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): | $34,3 : 1.000 = 0,03432 \text{ cm}/\mu\text{s}$ |



Der Ultraschallsensor soll als "Synthesizer" dienen: je nach Entfernung soll ein Ton gespielt werden.

Ein Beispiel:



Benötigte Bauteile:

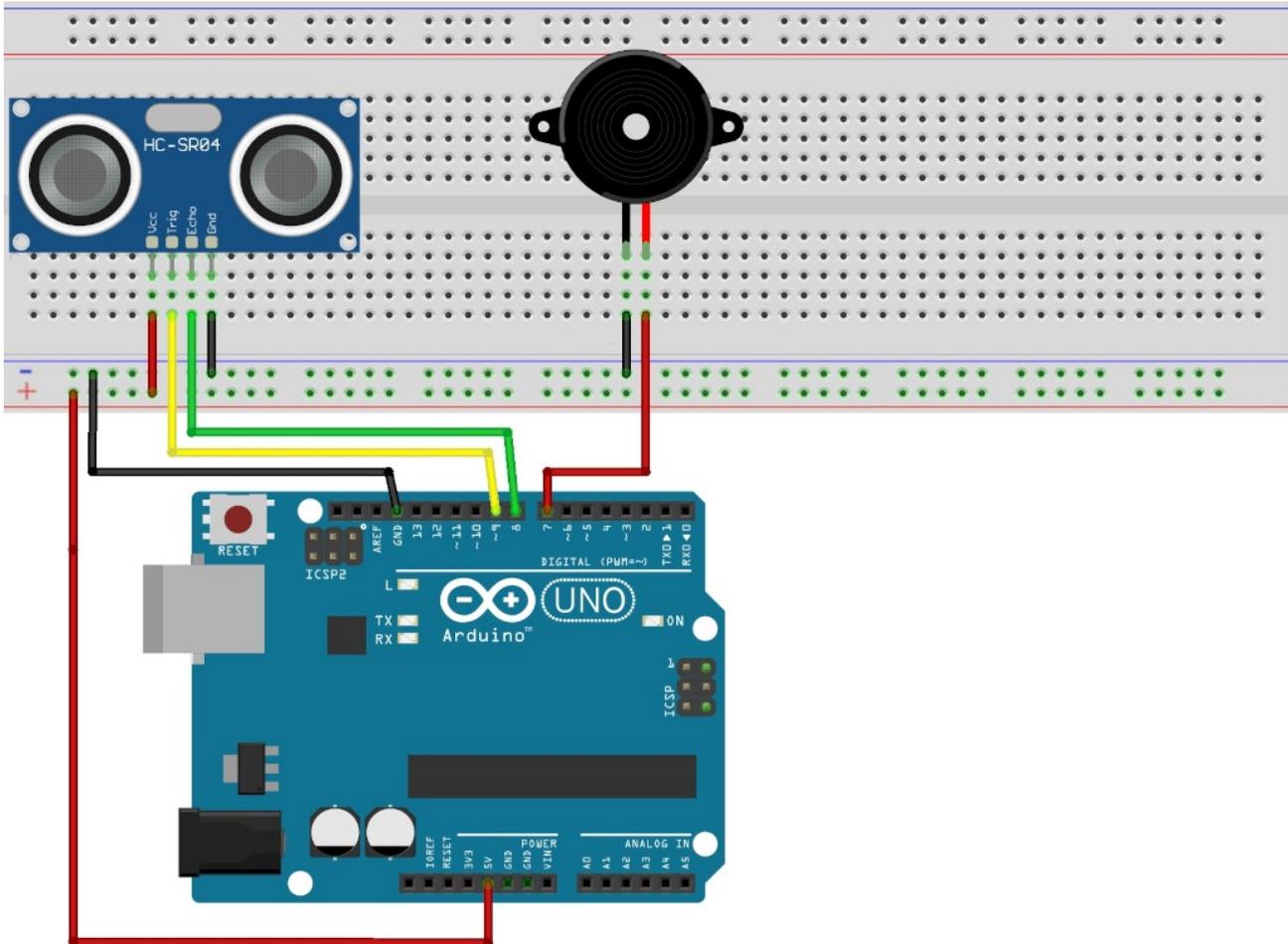
- ➔ Ultraschallsensor HC-SR04
- ➔ Lautsprecher
- ➔ Leitungsdrähte



Trig = 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.

Baue die Schaltung auf.



fritzing

Definiere die Variablen:

```
# define SENDEN 9
# define ECHO 8
# define LAUTSPRECHER 7

// Variable für Zeit und Entfernung initialisieren
long Zeit;
long Entfernung;
```

Der setup-Teil:

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

Es ist sinnvoll, das Abspielen des Tons nur unterhalb einer bestimmten Entfernung (hier 20 cm) zuzulassen.

Im loop-Teil setzt eine map-Anweisung die Entfernung in die Frequenz des Tons um:

```
void loop()
{
  // Sender kurz ausschalten um Störungen des Signals zu vermeiden
  digitalWrite(SENDEN, LOW);
  delay(10);

  // Signal für 10 Mikrosekunden senden
  digitalWrite(SENDEN, HIGH);
  delayMicroseconds(10);

  // Sender ausschalten
  digitalWrite(SENDEN, LOW);

  // pulseIn → Zeit messen, bis das Signal zurückkommt
  Zeit = pulseIn(ECHO, HIGH);

  /*
   Entfernung in cm berechnen
   Zeit/2 → nur eine Strecke soll berechnet werden
   Umrechnung in cm
  */
  Entfernung = (Zeit / 2) * 0.03432;

  // nur Entfernungen < 20 cm zulassen
  if (Entfernung < 20)
  {
    delay(500);

    // Entfernung 1 bis 20 auf Frequenzbereich 200 - 1000 umwandeln
    tone(LAUTSPRECHER, map(Entfernung, 1, 20, 200, 1000), 500);
  }
}
```

Du kannst natürlich auch andere Frequenzbereiche verwenden.

Hartmut Waller (hartmut-waller.info/arduino-blog) Letzte Änderung: 30.01.23