

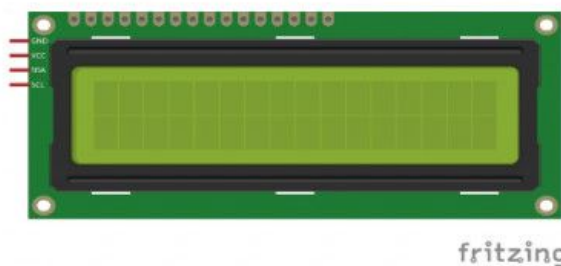
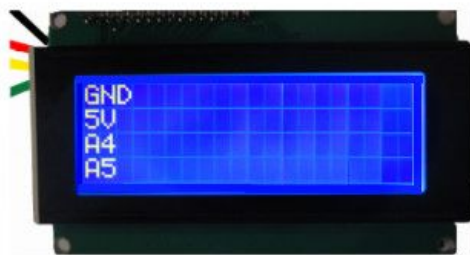
Entfernungen auf dem LCD anzeigen

Ein Ultraschallsensor misst die Entfernung zu einem Objekt und zeigt die Entfernung auf einem LCD an.

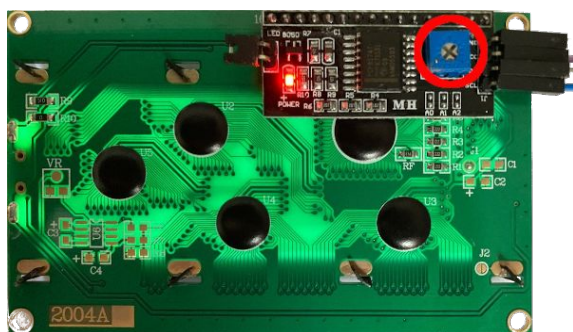
So sieht es aus:



SchlieÙe das LCD an:



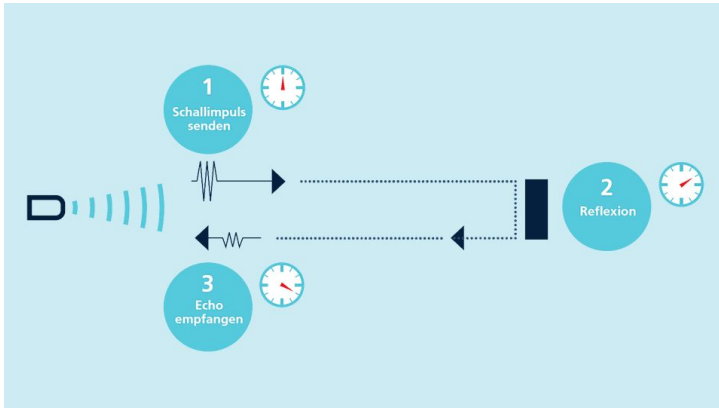
Normalerweise wäre eine komplexe Verkabelung zum Betrieb eines LCDs nötig. Der I2C-Bus regelt über einen eigenen Mikroprozessor die Kommunikation der Datenleitungen untereinander. Es werden deshalb nur vier Anschlüsse benötigt.



Die Helligkeit kann mit einem Potentiometer auf der Rückseite des LCDs eingestellt 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.

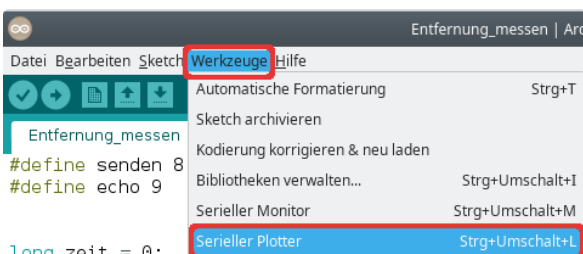
<https://www.microsonic.de/de/service/ultraschallsensoren/prinzip.htm>

Der Sender schickt das Signal und es wird vom Echo empfangen. Die Geschwindigkeit des Schalls in der Luft beträgt 343,2 m/s. Der Arduino bestimmt die Zeit in Millisekunden (1 Sekunde = 1000 Millisekunden), deshalb muss der Wert in cm pro Millisekunde berechnet werden:

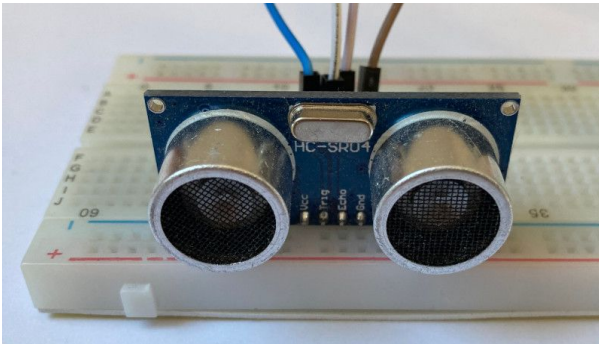
Umrechnung in cm:	$343,2 \text{ m} \cdot 100 = 34.320 \text{ cm}$
Strecke pro Sekunde:	$343.000 : 1.000 = 34,32 \text{ cm}$
Strecke pro Millisekunde:	$34,3 : 1.000 = 0,03432 \text{ cm}$



Der Arduino bietet nicht nur die Textausgabe mit Serial.print, er kann auch Ereignisse oder Messwerte mit einem Plotter aufzeichnen.



Den Seriellen Plotter findest du unter Werkzeuge → Serieller Plotter.

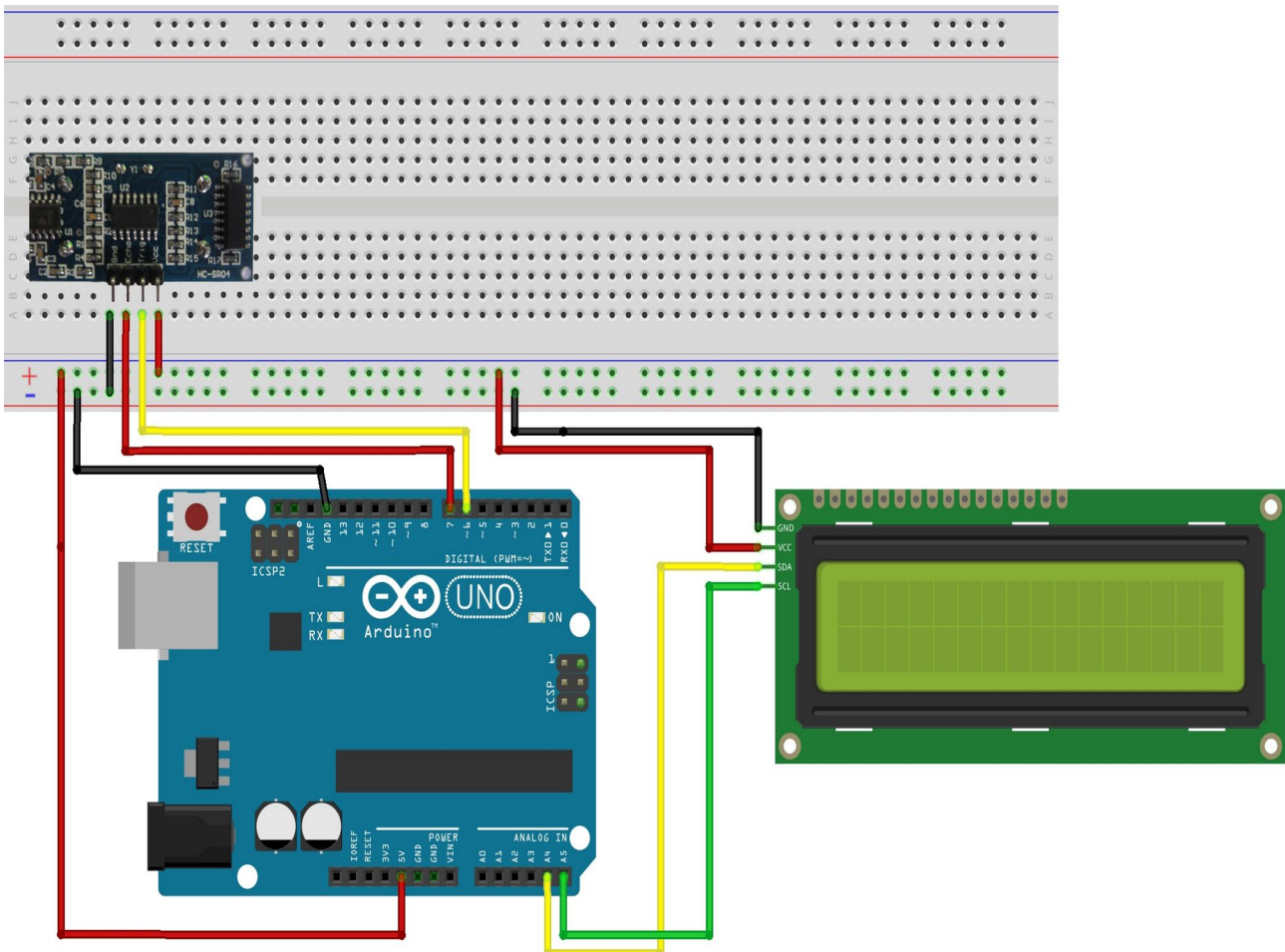


Damit der Ultraschallsensor ordnungsgemäß arbeiten kann, musst du ihn so einstecken, dass Sender und Empfänger nach vorn zeigen, damit das Signal ungehindert gesendet und empfangen werden kann.

Benötigte Bauteile:

- ➔ Ultraschallsensor HC-SR04
- ➔ LCD-Display I2C 1602
- ➔ Leitungsdrähte

Baue die Schaltung auf.



fritzing

Das Beispielprogramm zeigt die Entfernung im Seriellen Monitor/Seriellen Plotter an. Beachte die Kommentare.

```
// statt
// int SENDEN = 6;
// int ECHO = 7;
# define SENDEN 6
# define ECHO 7

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

void setup()
{
  Serial.begin(9600);
  pinMode(SENDEN, OUTPUT);
  pinMode(ECHO, INPUT);
}

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 Streckesoll berechnet werden
   Umrechnung in cm
  */
  Entfernung = (Zeit / 2) * 0.03432;

  // 400 cm ist die maximal messbare Entfernung
  if (Entfernung < 400)
  {
    // Ausgabe auf dem Seriellen Monitor
    Serial.print ("Entfernung: ");
    Serial.print(Entfernung);
    Serial.println(" cm");
    delay(1000);
  }
}
```

Ergänze das Programm für die Ausgabe auf dem LCD:

Benötigte Bibliothek:

Sketch → **Bibliothek einbinden** → **Bibliotheken verwalten**



Im Kopf des Programms wird das LCD definiert:

```
# include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

# define SENDEN 6
# define ECHO 7

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

Im setup-Teil wird das LCD gestartet ...

```
void setup()
{
  pinMode(SENDEN, OUTPUT);
  pinMode(ECHO, INPUT);
  lcd.init();
  lcd.backlight();
}
```

... und der loop-Teil wird durch die Ausgabe auf dem LCD ergänzt.

```
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;
// 400 cm ist die maximal messbare Entfernung
if (Entfernung < 400)
{
  // Ausgabe auf dem LCD
  lcd.setCursor(0, 0);
  lcd.print("Entfernung messen");
  lcd.setCursor(0, 1);
  lcd.print("-----");
  lcd.setCursor(0, 2);
  lcd.print("Entfernung: ");
  lcd.print(Entfernung);
  lcd.print(" cm  ");
  delay(1000);
}
}
```