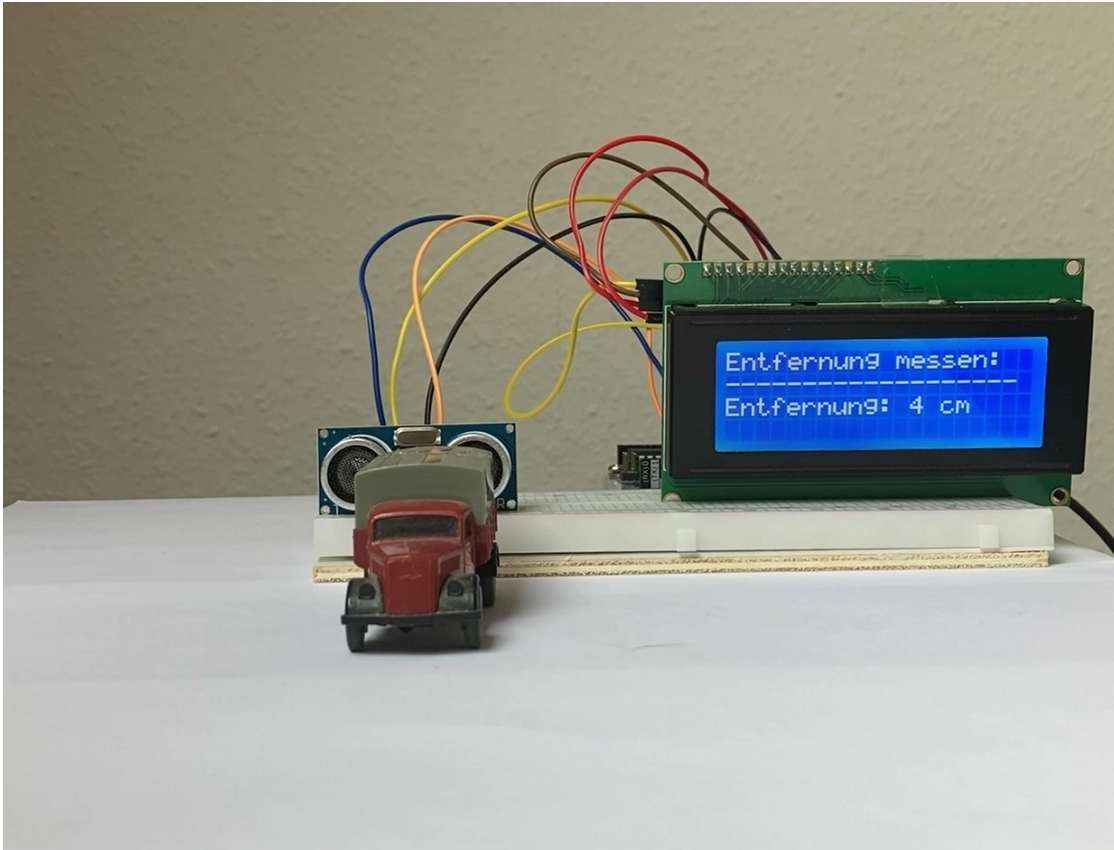
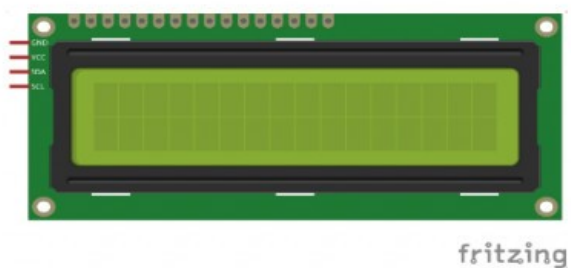


Ein Ultraschallsensor misst die Entfernung zu einem Objekt und zeigt die Entfernung auf einem LCD an.

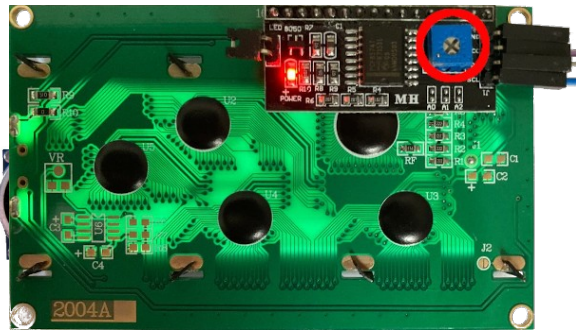
So sieht es aus:



SchlieÙe das LCD an:

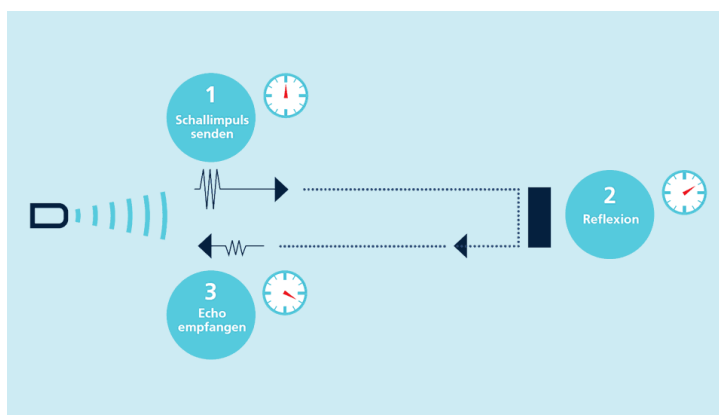


Normalerweise wäre eine komplexe Verkabelung zum Betrieb eines LCDs nötig. Der I2C-Bus regelt über einen eigenen Mikroprozessor die Kommunikation der Datenleitungen untereinander. Es werden deshalb nur vier Anschlüsse benötigt.



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>

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

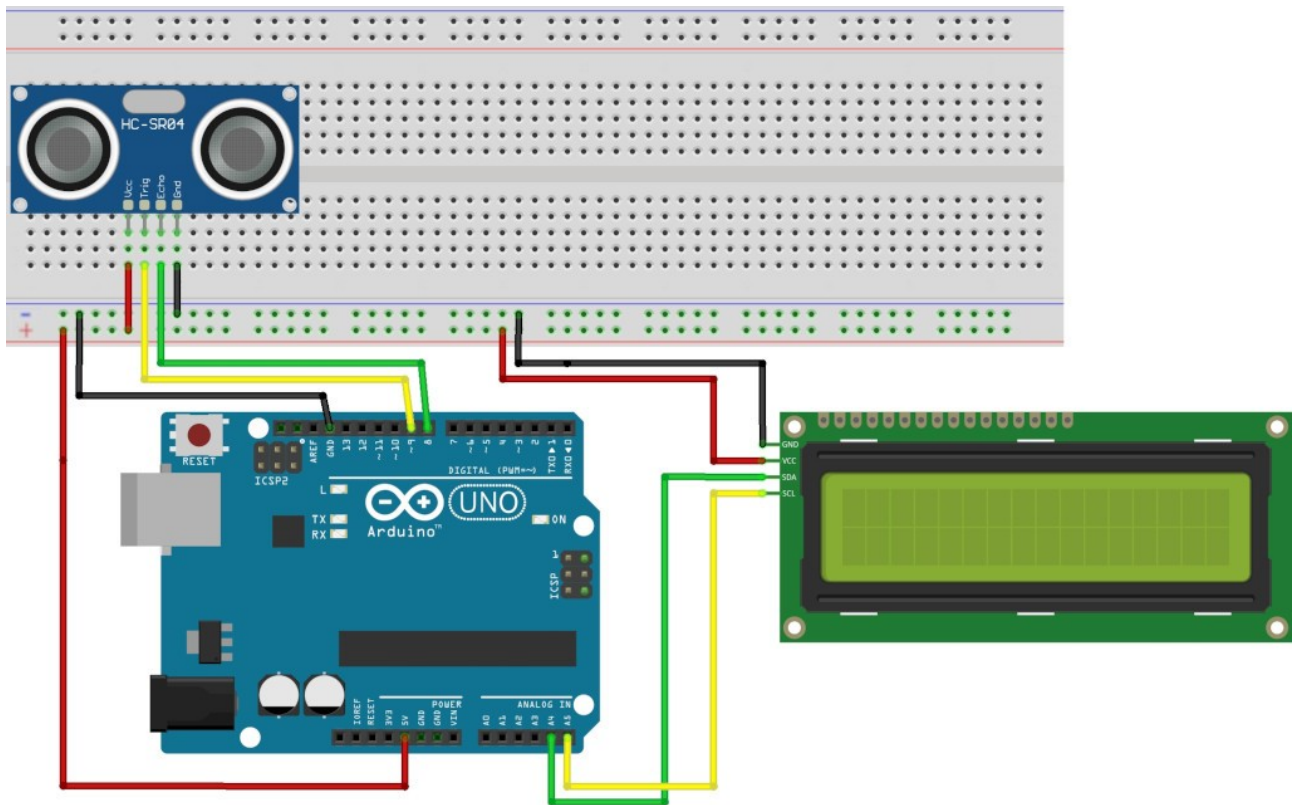


Damit der Ultraschallsensor ordnungsgemäß arbeiten kann, musst du ihn so einstecken, dass Sender und Empfänger nach vorn zeigen, damit das Signal ungehindert gesendet und empfangen werden kann.

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.

```
# define SENDEN 7
# define ECHO 6

// 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);

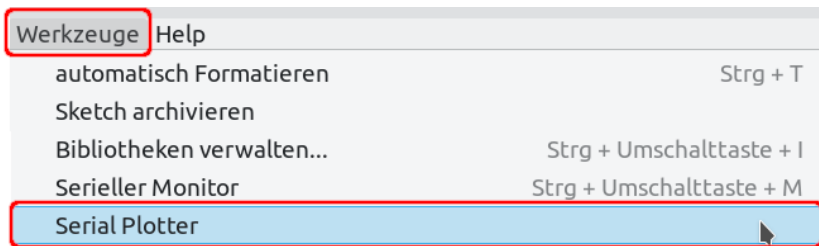
    /* Entfernung in cm berechnen
       Zeit/2 → nur eine Strecke soll berechnet werden
       Umrechnung in cm
    */
    Entfernung = (Zeit / 2) * 0.03432;

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

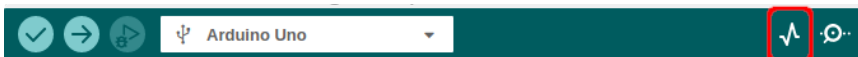
So sieht es im Seriellen Plotter aus:



Den Seriellen Plotter findest du im Menü ...

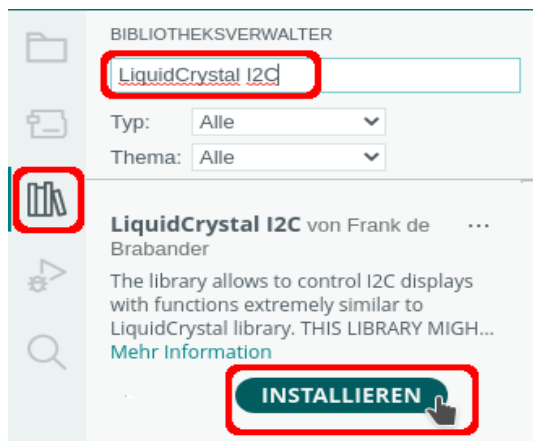


... oder als Symbol in der Statusleiste.



Ergänze das Programm für die Ausgabe auf dem LCD:

Benötigte Bibliothek:



Im Kopf des Programms wird die Bibliothek eingebunden und die Parameter (Adresse, Anzahl der Spalten und Anzahl der Zeilen) für das LCD definiert.

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

# define SENDEN 7
# define ECHO 6

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

Im setup-Teil wird das LCD gestartet und der Modus für Senden und Empfangen definiert ...

```
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);
  }
}
```