

# JOY-IT

Joy-IT<sup>®</sup> Ultraschall Abstandssensor



## Index

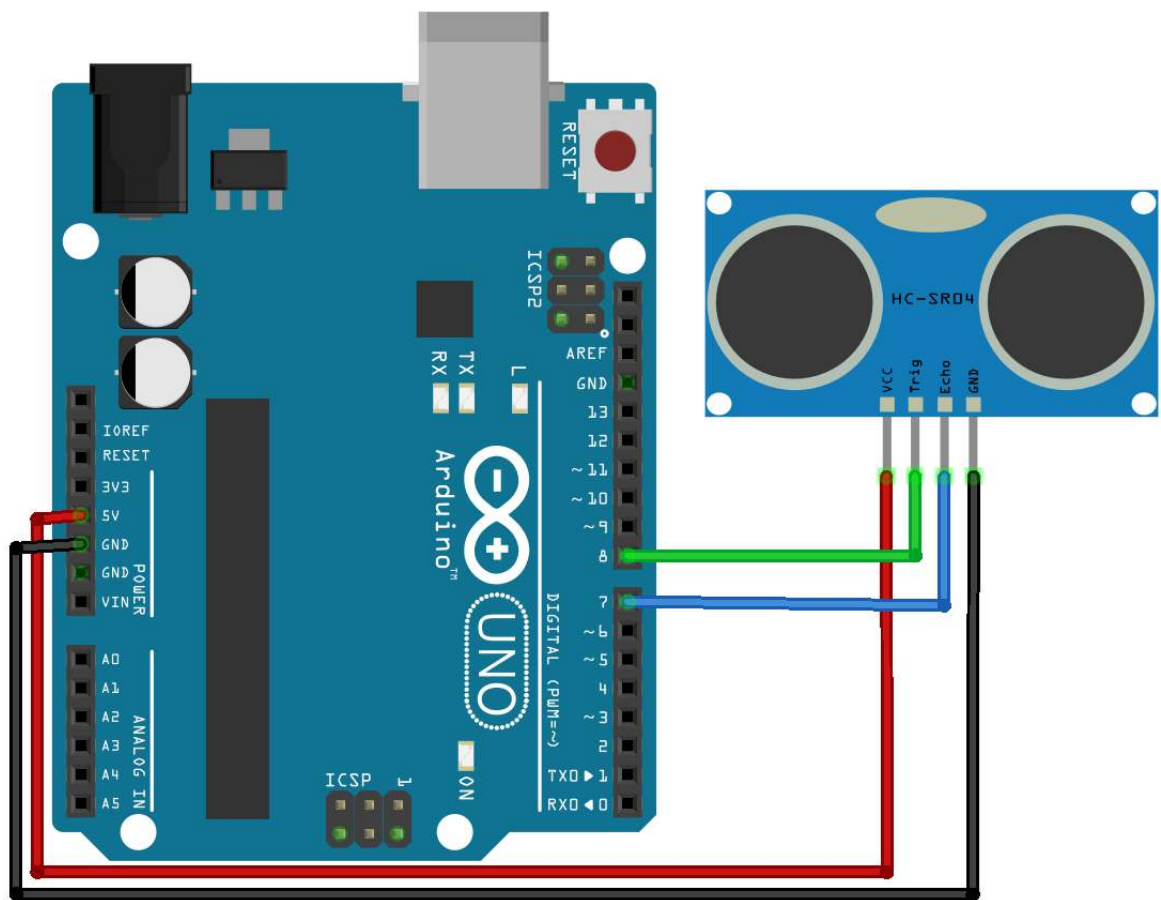
1. Verwendung mit einem Arduino
  - 1.1 Anschluss
  - 1.2 Code-Beispiel
  
2. Verwendung mit einem Raspberry Pi
  - 2.1 Installation des System
  - 2.2 Anschluss
  - 2.3 Code-Beispiel
  
3. Support

Sehr geehrter Kunde,  
 vielen Dank, dass Sie sich für unser Produkt entschieden haben. Im Folgenden haben wir aufgelistet, was bei der Inbetriebnahme zu beachten ist.

# 1. Verwendung mit einem Arduino

## 1.1 Anschluss

Schließen Sie das Ultraschall Modul, wie im folgenden Bild, bzw. in der folgenden Tabelle, zu sehen, an Ihren Arduino an.



Arduino	Ultraschall Abstandssensor
5V	VCC
GND	GND
7	Echo
8	Trig

## 1.2 Code-Beispiel

Nachfolgend können Sie ein Code-Beispiel entnehmen, welches die Funktionalität des Sensors demonstriert.

Übertragen Sie den Code bitte vollständig auf Ihren Arduino.

```
#define Echo_EingangPin 7 // Echo Eingangs-Pin
#define Trigger_AusgangPin 8 // Trigger Ausgangs-Pin

// Benoetigte Variablen werden definiert
int maximumRange = 300;
int minimumRange = 2;
long Abstand;
long Dauer;

void setup() {
  pinMode(Trigger_AusgangPin, OUTPUT);
  pinMode(Echo_EingangPin, INPUT);
  Serial.begin (9600);
}

void loop() {
  // Abstandsmessung wird mittels des 10us langen Triggersignals gestartet
  digitalWrite(Trigger_AusgangPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trigger_AusgangPin, LOW);

  // Nun wird am Echo-Eingang gewartet, bis das Signal aktiviert wurde
  // und danach die Zeit gemessen, wie lang es aktiviert bleibt
  Dauer = pulseIn(Echo_EingangPin, HIGH);

  // Nun wird der Abstand mittels der aufgenommenen Zeit berechnet
  Abstand = Dauer/58.2;
  // Überprüfung ob gemessener Wert innerhalb der zulässigen Entfernung liegt
  if (Abstand >= maximumRange || Abstand <= minimumRange) {
    // Falls nicht wird eine Fehlermeldung ausgegeben.
    Serial.println("Abstand ausserhalb des Messbereichs");
    Serial.println("-----");
  }

  else {
    // Der berechnete Abstand wird in der seriellen Ausgabe ausgegeben
    Serial.print("Der Abstand betraegt:");
    Serial.print(Abstand);
    Serial.println("cm");
    Serial.println("-----");
  }
  // Pause zwischen den einzelnen Messungen
  delay(500);
}
```

## 2. Verwendung mit einem Raspberry Pi

### 2.1 Installation des Systems

Wenn Sie auf Ihrem Raspberry Pi bereits die aktuellste Raspbian Software verwenden, können Sie diesen Schritt überspringen und sofort mit dem nächsten Schritt fortfahren. Andernfalls folgen Sie bitte den weiteren Anweisungen.

Installieren Sie auf Ihre SD-Karte bitte das aktuellste Raspbian System-Image. Dieses können Sie [hier](#) herunterladen.

Mit einem geeigneten Programm (z.B. Etcher) können Sie das Image dann auf Ihre SD Karte übertragen.



Nach erfolgreicher Übertragung kann die SD Karte in den Raspberry Pi eingelegt und das System gestartet werden.

## 2.2 Anschluss

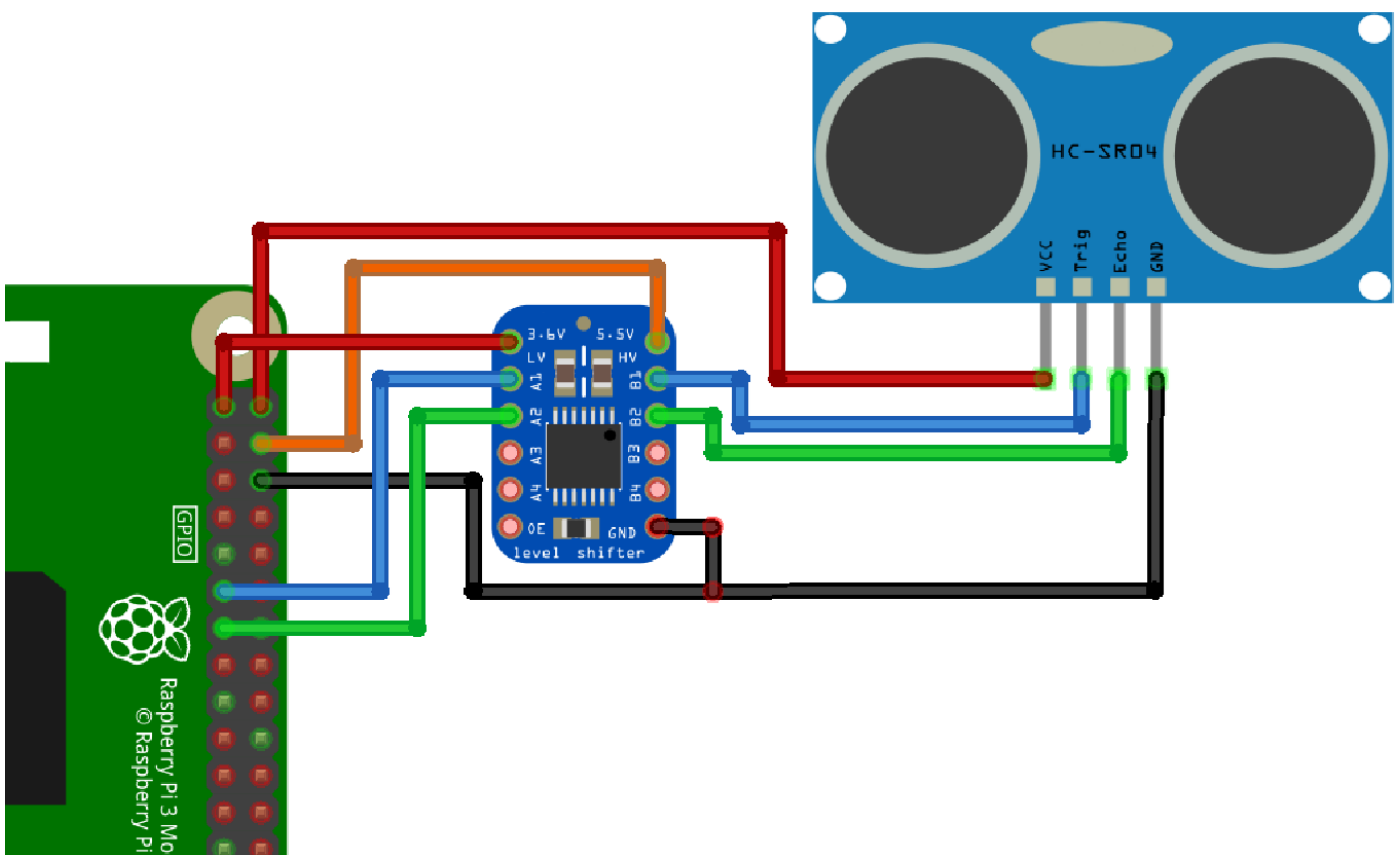
Schließen Sie das Ultraschall Modul, wie im folgenden Bild, bzw. in der folgenden Tabelle, zu sehen, an Ihren Raspberry Pi an.

**Achtung!** Der Raspberry Pi arbeitet mit seinem ARM-Prozessorkern mit einem Spannungslevel von 3,3V. Der Ultraschall Abstandssensor funktioniert jedoch nur mit dem höheren Spannungslevel.

Würde man diesen Sensor uneingeschränkt am Raspberry Pi, ohne Vorsichtsmaßnahmen, betreiben, könnte dies zu permanenten Schäden an den Eingängen des Raspberry Pis führen.

Aus diesem Grund sollte für die Verwendung des Sensors an einem Raspberry Pi unbedingt ein Voltage Translator verwendet werden, der das Spannungslevel anpasst und einen sicheren Betrieb gewährleistet.

Hierfür kann z.B. der KY-051 Voltage Translator aus unserem SensorKit X40 verwendet werden. Im nachfolgenden wird die Vorgehensweise unter Gebrauch des KY-051 beschrieben.



Abstandssensor	Voltage Translator
VCC	
Trig	B1
Echo	B2
GND	GND

Abstandssensor	Raspberry Pi
VCC	+5V (Pin 2)

Voltage Translator	Raspberry Pi
5.5V	5V (Pin 3)
3.6V	3.3V (Pin 1)
A1	GPIO17 (Pin 11)
A2	GPIO27 (Pin 13)
GND	GND (Pin 6)

## 1.2 Code-Beispiel

Nachfolgend können Sie ein Code-Beispiel entnehmen, welches die Funktionalität des Sensors demonstriert.

Übertragen Sie den Code bitte vollständig auf Ihren Raspberry Pi.



```

# coding=utf-8
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

# Hier können die jeweiligen Eingangs-/Ausgangspins ausgewählt werden
Trigger_AusgangsPin = 17
Echo_EingangsPin    = 27

# Pause zwischen den einzelnen Messungen in Sekunden
sleeptime = 0.8

GPIO.setup(Trigger_AusgangsPin, GPIO.OUT)
GPIO.setup(Echo_EingangsPin, GPIO.IN)
GPIO.output(Trigger_AusgangsPin, False)

# Hauptprogrammschleife
try:
    while True:
        # Abstandsmessung mittels des 10us langen Triggersignals
        GPIO.output(Trigger_AusgangsPin, True)
        time.sleep(0.00001)
        GPIO.output(Trigger_AusgangsPin, False)

        # Hier wird die Stopuhr gestartet
        EinschaltZeit = time.time()
        while GPIO.input(Echo_EingangsPin) == 0:
            EinschaltZeit = time.time()
        while GPIO.input(Echo_EingangsPin) == 1:
            AusschaltZeit = time.time()

        # Die Differenz der beiden Zeiten ergibt die gesuchte Dauer
        Dauer = AusschaltZeit - EinschaltZeit
        # Berechne Abstand auf Basis der Schallgeschwindigkeit
        Abstand = (Dauer * 34300) / 2

        # Überprüfung, ob gemessener Wert innerhalb der zulässigen Entfernung
        if Abstand < 2 or (round(Abstand) > 300):
            # Falls nicht wird eine Fehlermeldung ausgegeben
            print("Abstand außerhalb des Messbereich")
            print("-----")
    
```

```
else:
    # Der Abstand wird auf zwei Stellen hinterm Komma formatiert
    Abstand = format((Dauer * 34300) / 2, '.2f')
    # Der berechnete Abstand wird auf der Konsole ausgegeben
    print("Der Abstand beträgt:"), Abstand, ("cm")
    print("-----")

    # Pause zwischen den einzelnen Messungen
    time.sleep(sleeptime)

# Aufraeumarbeiten nachdem das Programm beendet wurde
except KeyboardInterrupt:
    GPIO.cleanup()
```

### 3. Support

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: [service@joy-it.net](mailto:service@joy-it.net)

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 98469 – 66 (11- 18 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

[www.joy-it.net](http://www.joy-it.net)