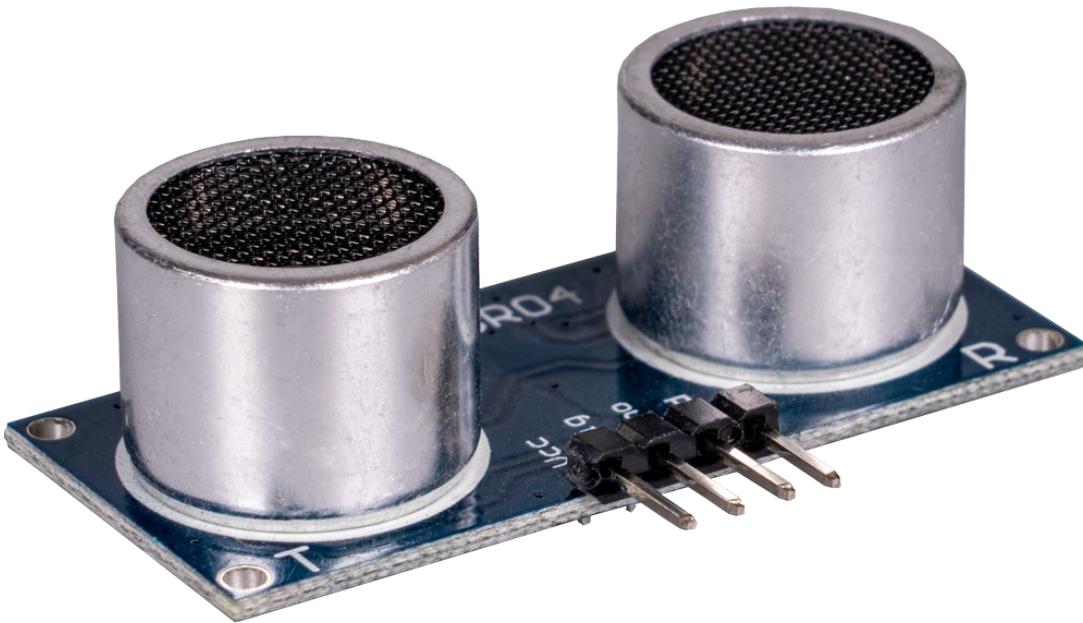


ULTRASCHALL-ABSTANDSSENSOR

SEN-US01



1. ALLGEMEINE INFORMATIONEN

Sehr geehrte*r Kunde *in,
vielen Dank, dass Sie sich für unser Produkt entschieden haben. Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist.

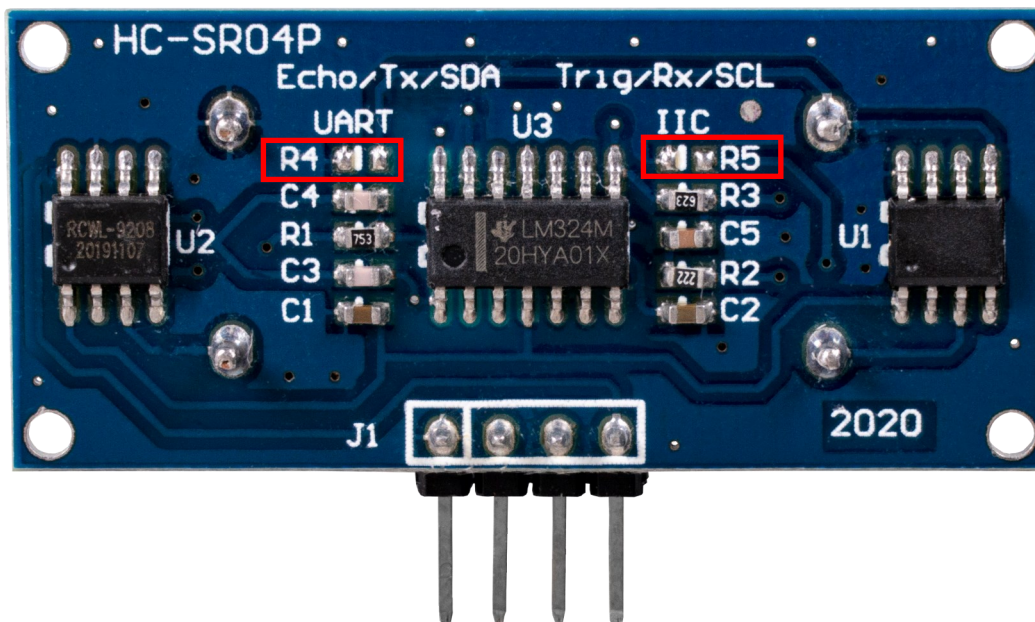
Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

Dieses Ultraschallmodul kann Entfernungen von 2 bis 450 cm, mit einer Auflösung von 1 mm messen.

Dies geschieht mit Hilfe von Ultraschall. Dabei wird ein Ultraschallsignal ausgegeben, welches auf ein Hindernis trifft. Dieses wird reflektiert und zum Sensor zurückgeführt. Aus dem Zeitraum, indem das Signal unterwegs war und der Schallgeschwindigkeit, mit der sich das Signal fortbewegt, lässt sich die Entfernung des Hindernisses berechnen.

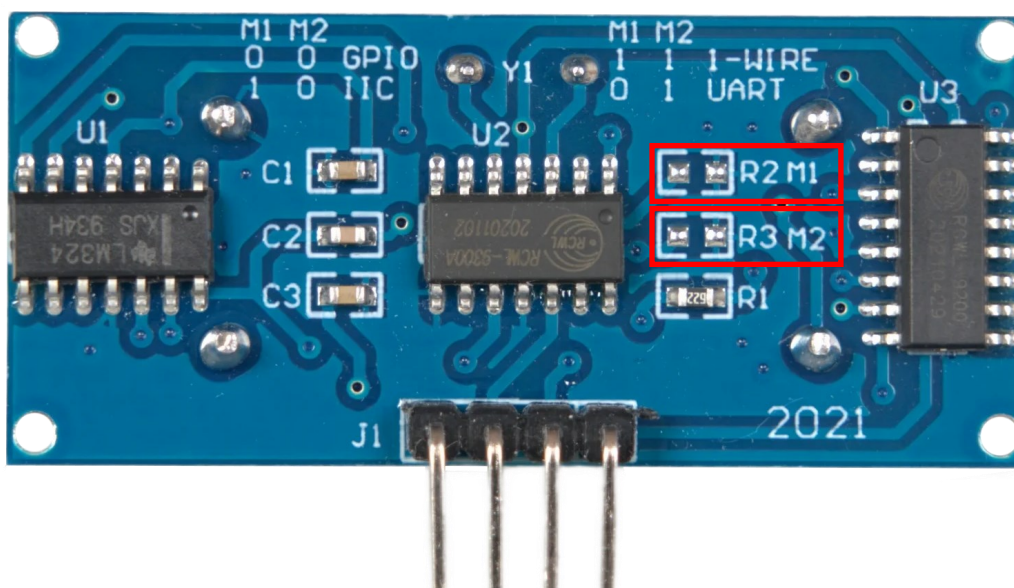
2. SCHNITTSTELLEN

Der SEN-US01 besitzt die drei Schnittstellen GPIO, UART und I2C. Die gewünschte Schnittstelle kann durch das auflöten eines Widerstandes ausgewählt werden. Dafür sind die Löt pads R4 und R5 entscheidend. Diese müssen mittels eines 10k Ω Widerstand (0603) verbunden werden.



	GPIO	UART	I2C
R4	Nicht verbunden	Nicht verbunden	10k Ω
R5	Nicht verbunden	10k Ω	Nicht verbunden

Neue Version mit 1-Wire



	GPIO	UART	I2C	1-Wire
R2 M1	Nicht verbunden	Nicht verbunden	10k Ω	10k Ω
R3 M2	Nicht verbunden	10k Ω	Nicht verbunden	10k Ω

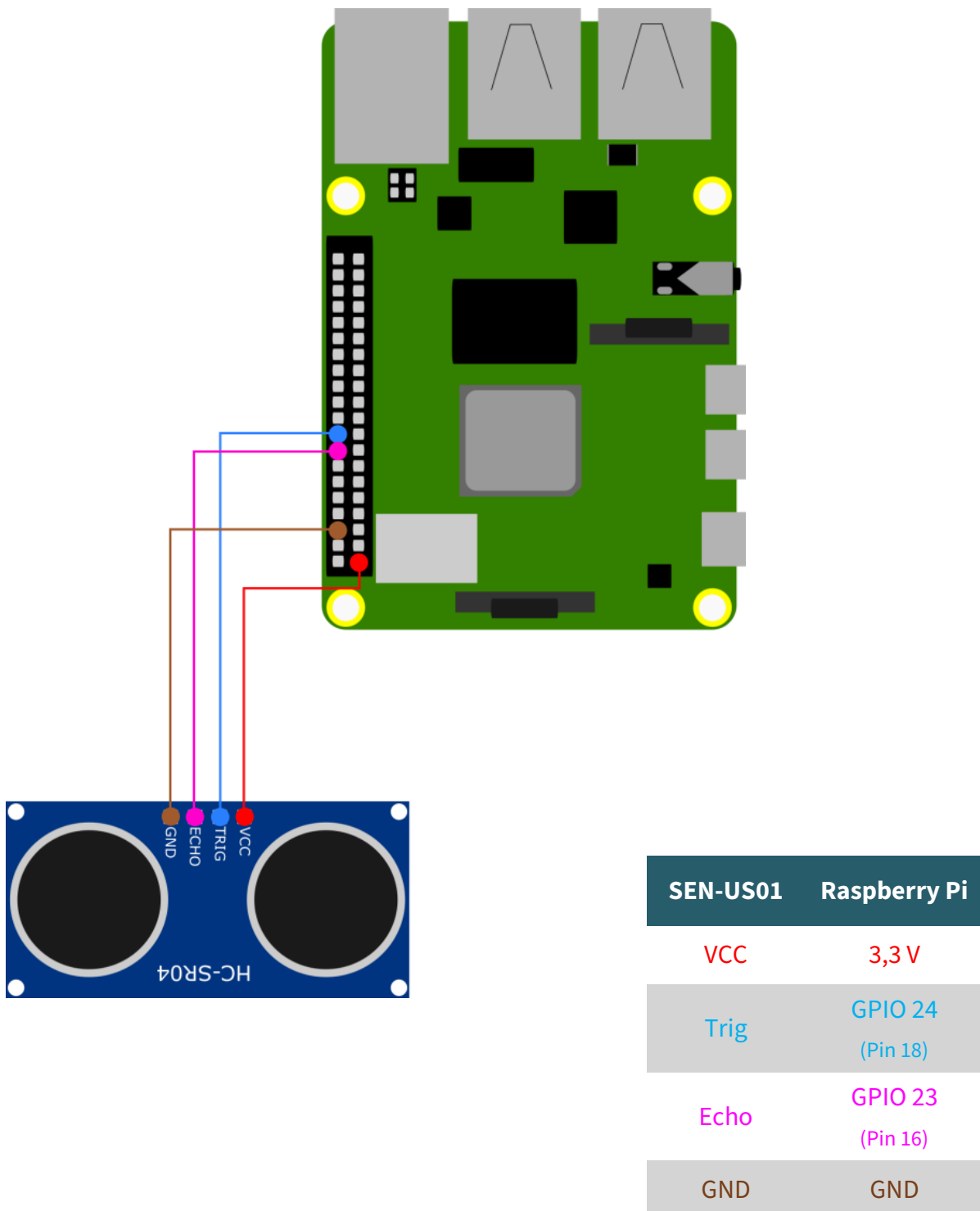
3. VERWENDUNG MIT DEM RASPBERRY PI



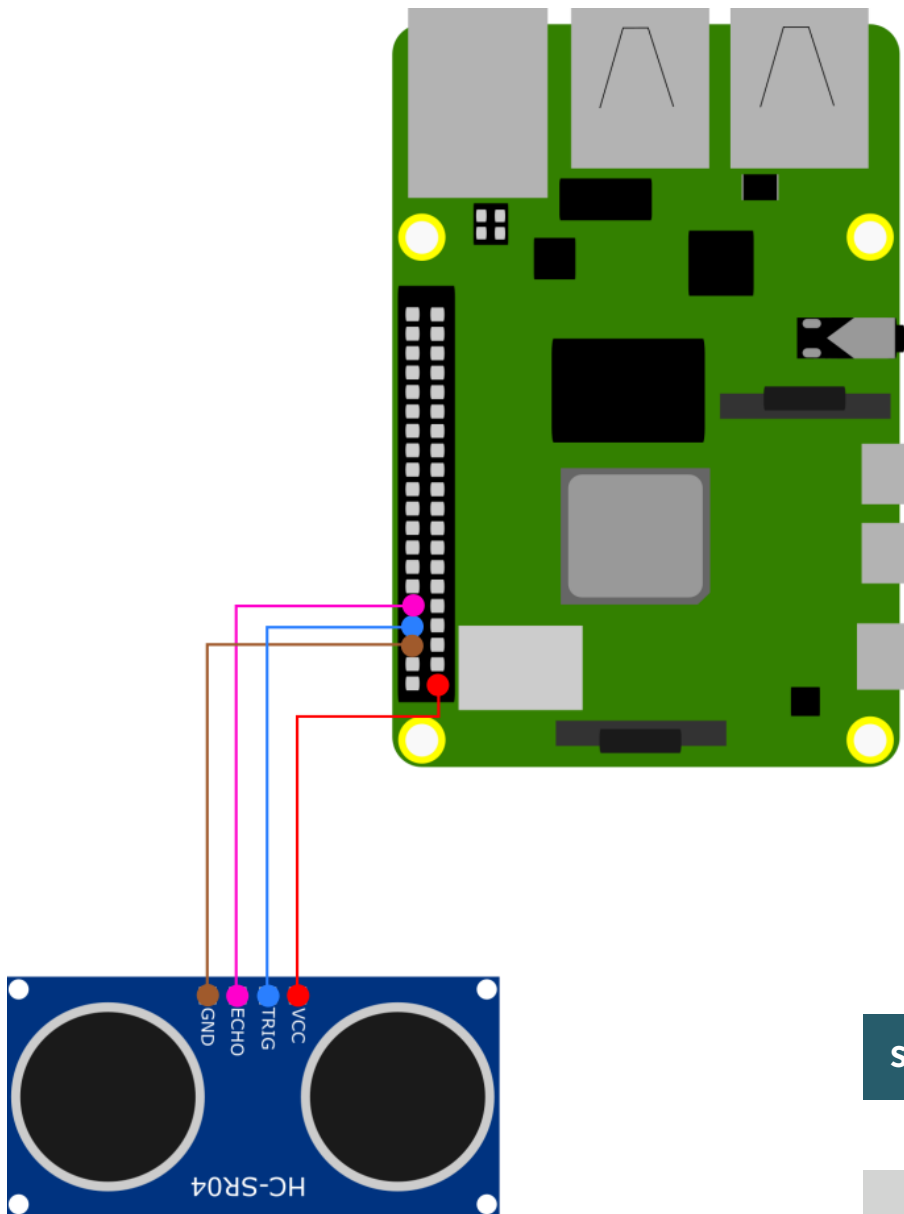
Diese Anleitung wurde unter Raspberry Pi OS Bookworm für den Raspberry Pi 4 und 5 geschrieben. Es wurde keine Überprüfung mit neueren Betriebssystemen oder Hardware durchgeführt.

3.1 Anschluss

3.1.1 Schnittstelle GPIO

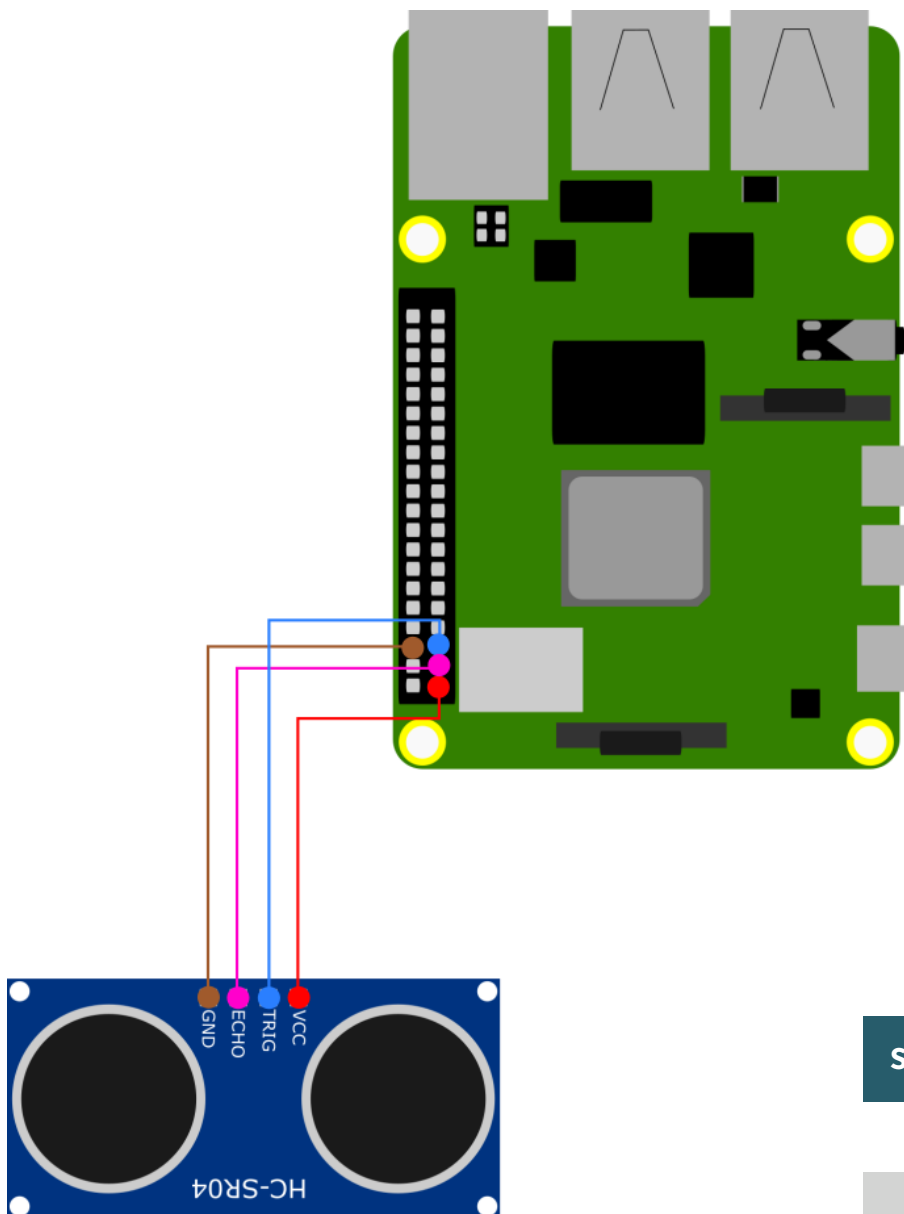


3.1.2 Schnittstelle UART



SEN-US01	Raspberry Pi
VCC	3,3 V
Trig / Rx	GPIO 14 (Pin 8 / TXD)
Echo / Tx	GPIO 15 (Pin 10 / RXD)
GND	GND

3.1.3 Schnittstelle I2C



SEN-US01	Raspberry Pi
VCC	3,3 V
Trig / SCL	GPIO 3 (Pin 5 / SCL)
Echo / SDA	GPIO 2 (Pin 3 / SDA)
GND	GND

3.2 Codebeispiel

Wir stellen Ihnen ein Codebeispiel zur Verfügung, welches Sie sich [hier](#) herunterladen können oder Sie geben den folgenden Befehl in Ihre Konsole ein.

```
wget https://www.joy-it.net/files/files/Produkte/SEN-US01/SEN-US01_Codeexample_RaspberryPi.zip
```

Entpacken Sie nun die .zip Datei mit dem folgenden Befehl:

```
unzip SEN-US01_Codeexample_RaspberryPi.zip
```

3.2.1 für die Schnittstelle GPIO

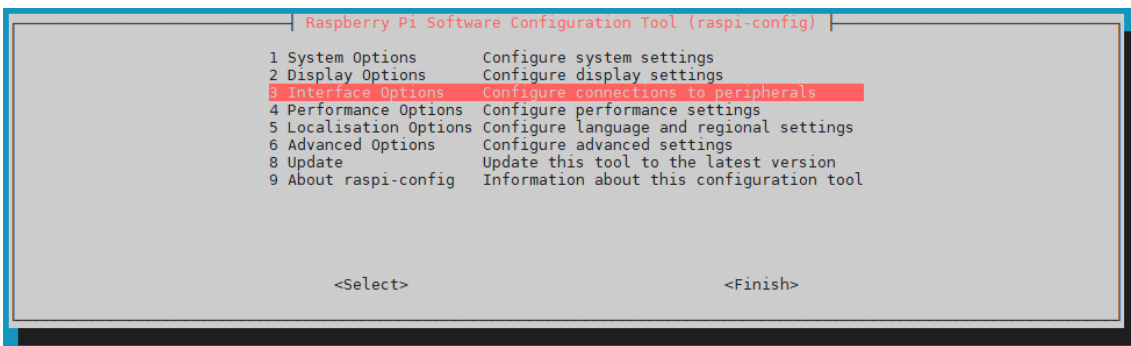
Für diesen Anschluss müssen Sie keine weiteren Installationen durchführen. Sie können also direkt das zugehörige Codebeispiel mit dem folgenden Befehl ausführen. Achten Sie darauf, dass Ihr Verzeichnis abweichen kann.

```
python3 SEN-US01_Codeexample_RaspberryPi/SEN-US01_GPIO.py
```

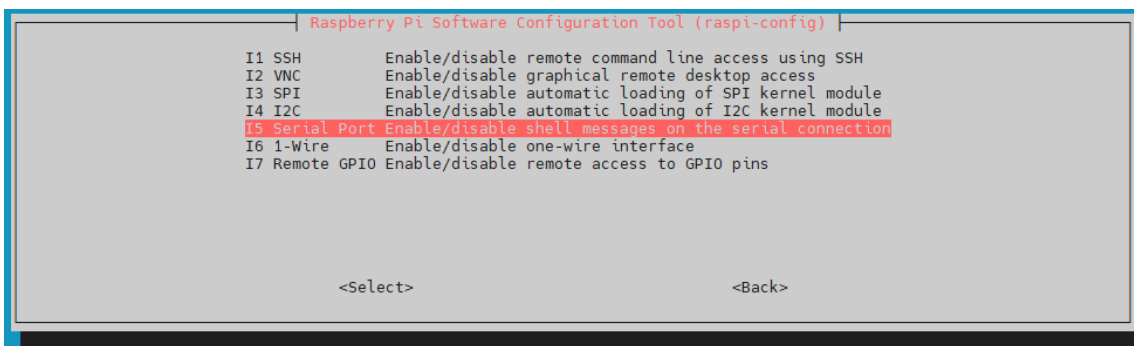
3.2.2 für die Schnittstelle UART

Zunächst müssen Sie in den Einstellungen die serielle Kommunikation aktivieren. Dafür geben Sie den folgenden Befehl in die Konsole ein.

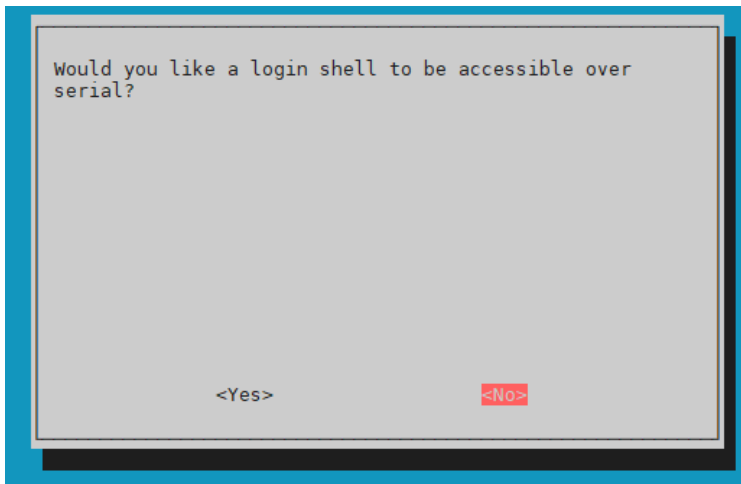
```
sudo raspi-config
```



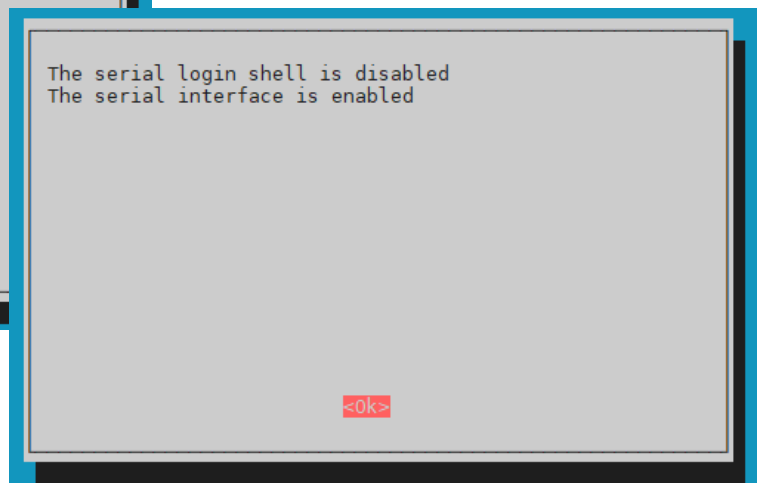
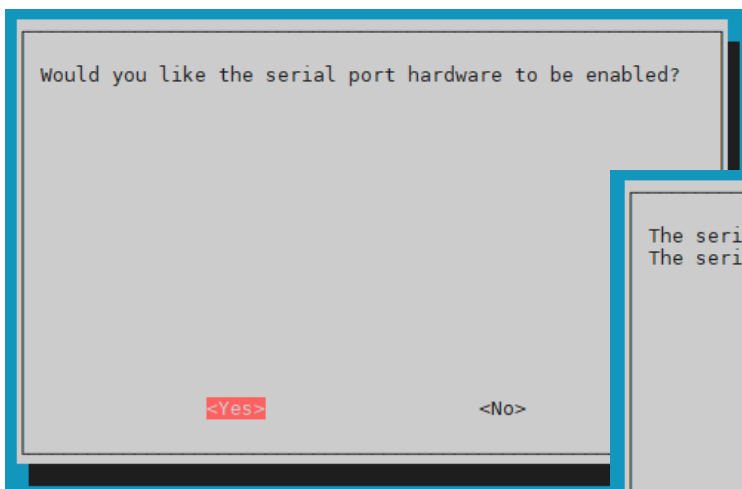
Wählen Sie dort unter **3 Interface Options** → **I5 Serial Port** aus.



Klicken Sie dort auf **No**, wenn Ihnen die Frage **Would you like a login shell to be accessible over serial?** gestellt wird.



Stimmen Sie der Frage **Would you like the serial port hardware to be enabled?** zu.



Starten Sie danach den Raspberry Pi mit dem folgenden Befehl neu.

```
sudo reboot
```

Führen Sie nun den folgenden Befehl aus, um die Bibliothek zur seriellen Kommunikation zu installieren.

```
sudo apt-get install python3-serial
```

Das verwendete Codebeispiel ist für den Raspberry Pi 5 konfiguriert, um den Pi4 zu nutzen, müssen Sie das #-Zeichen am Anfang von Zeile 6 entfernen und es am Anfang von Zeile 7 einfügen.

```
5 # Open serial port
6 ser = serial.Serial("/dev/ttyS0", 9600, timeout = 1) # Pi4
7 #ser = serial.Serial("/dev/ttyAMA0", 9600, timeout = 1) # Pi5
```

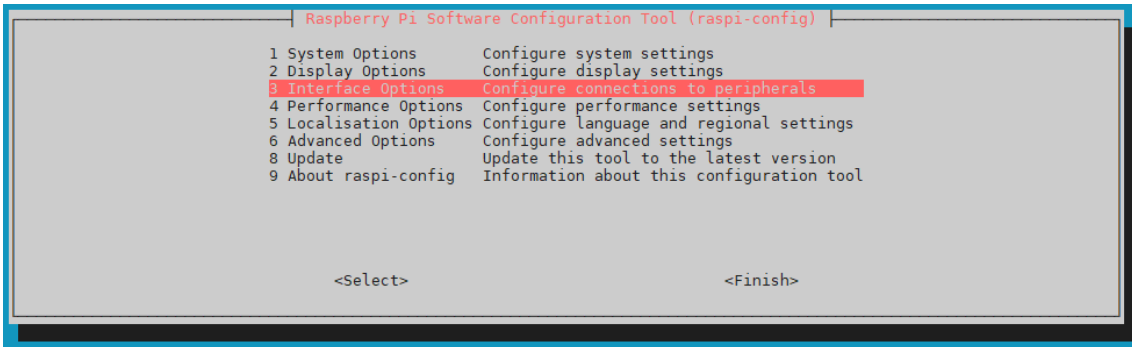
Sie können nun mit dem folgenden Befehl unseren Beispielcode zur UART-Schnittstelle ausführen. Achten Sie dabei darauf, dass Ihr Verzeichnis abweichen kann.

```
python3 SEN-US01_Codeexample_RaspberryPi/SEN-US01_UART.py
```

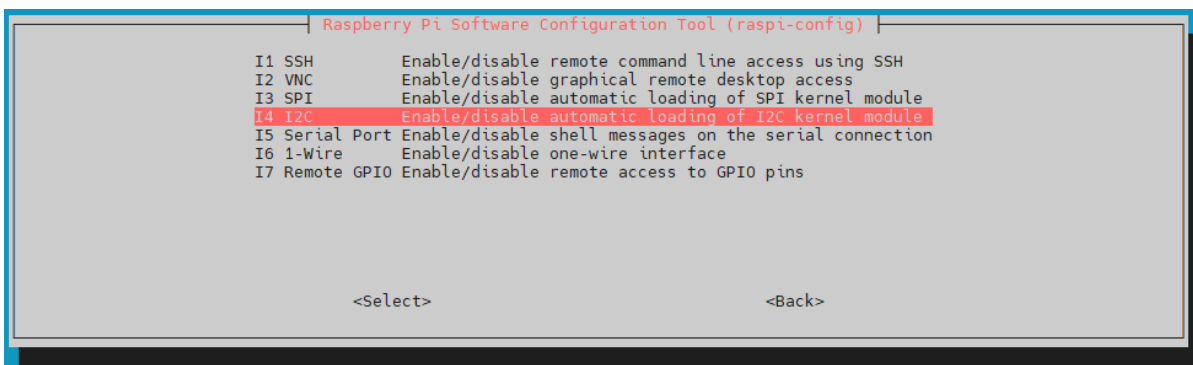
3.2.3 für die Schnittstelle I2C

Zunächst müssen Sie in den Einstellungen die serielle Kommunikation aktivieren. Dafür geben Sie den folgenden Befehl in die Konsole ein.

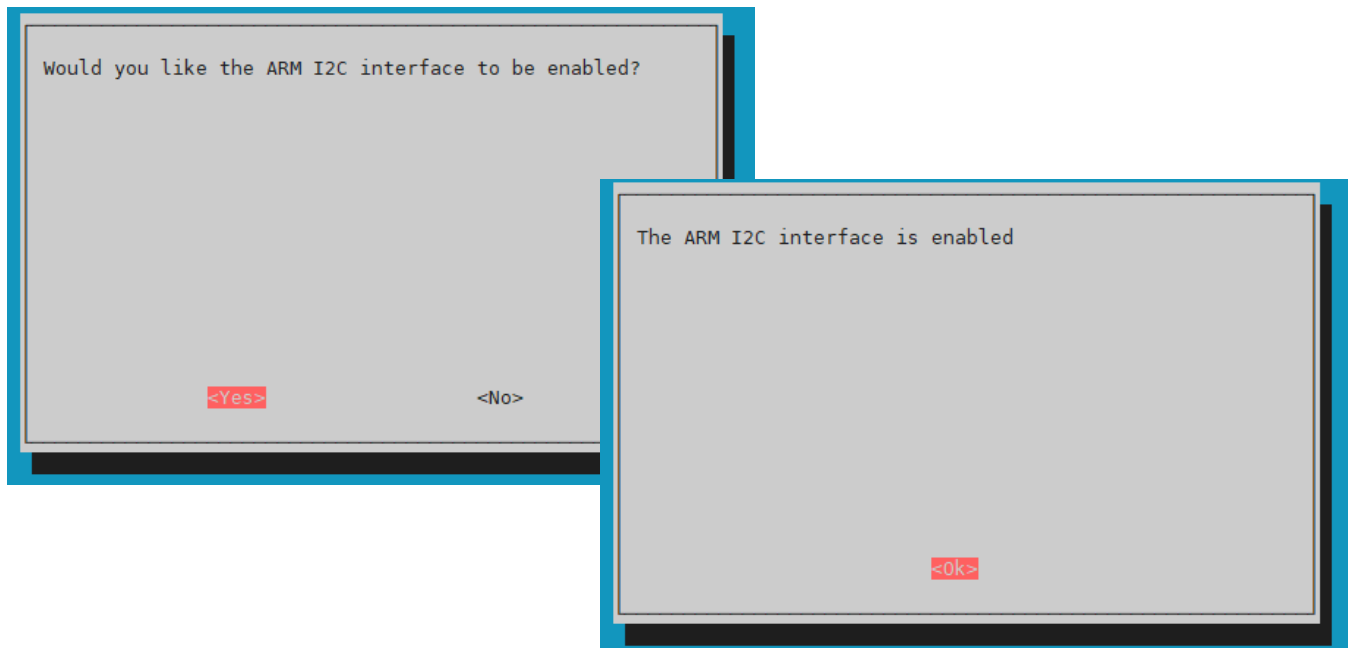
```
sudo raspi-config
```



Wählen Sie dort unter **3 Interface Options** → **I4 I2C** aus.



Klicken Sie dort auf **Yes**, wenn Ihnen die Frage **Would you like the ARM**



Installieren Sie nun mit dem folgenden zwei Befehlen die benötigten Bibliothek für die I2C-Kommunikation.

```
sudo apt-get install i2c-tools
```

```
sudo apt-get install python3-smbus
```

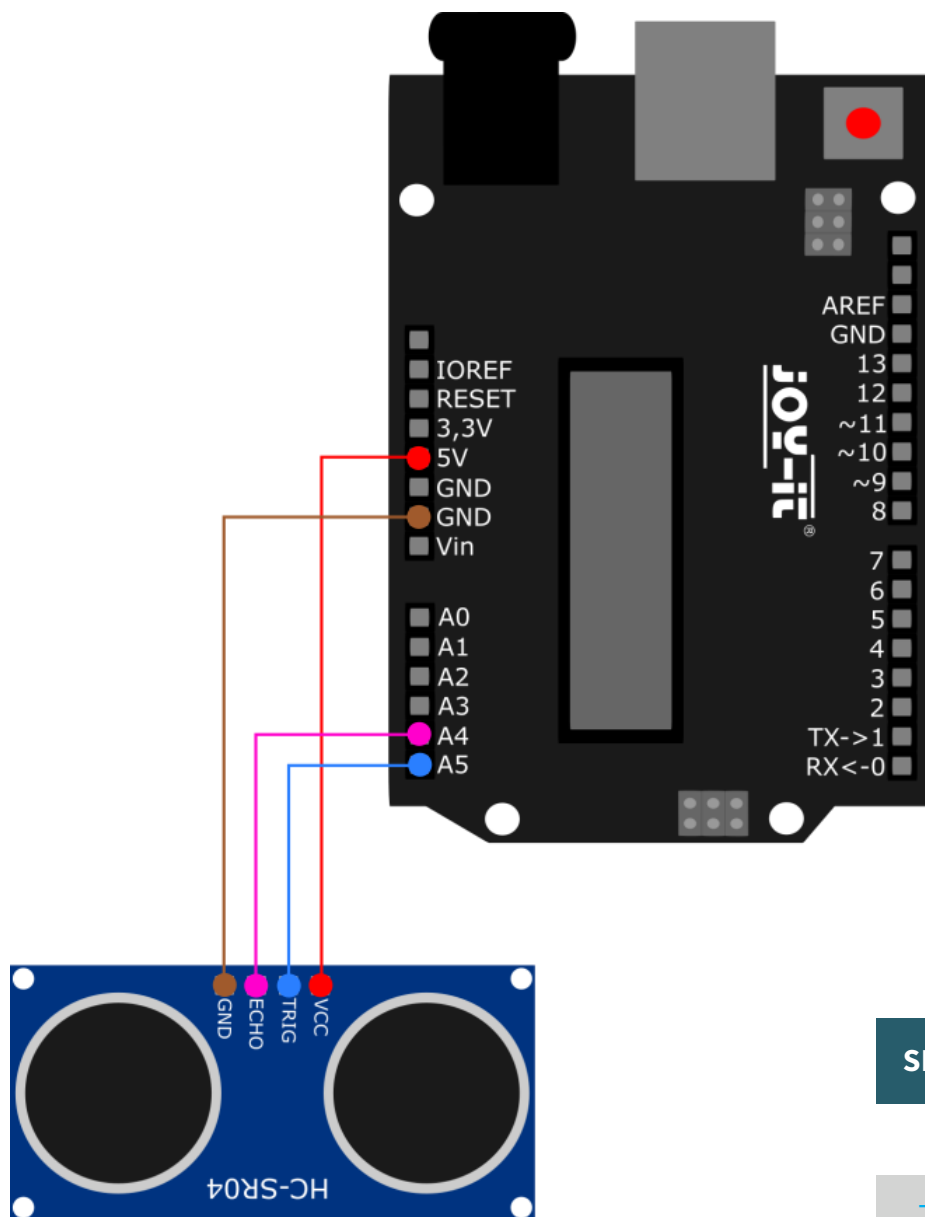

Sie können nun mit dem folgenden Befehl unseren Beispielcode zur I2C-Schnittstelle ausführen. Achten Sie dabei darauf, dass Ihr Verzeichnis abweichen kann.

```
python3 SEN-US01_Codeexample_RaspberryPi/SEN-US01_I2C.py
```

4. VERWENDUNG MIT DEM ARDUINO

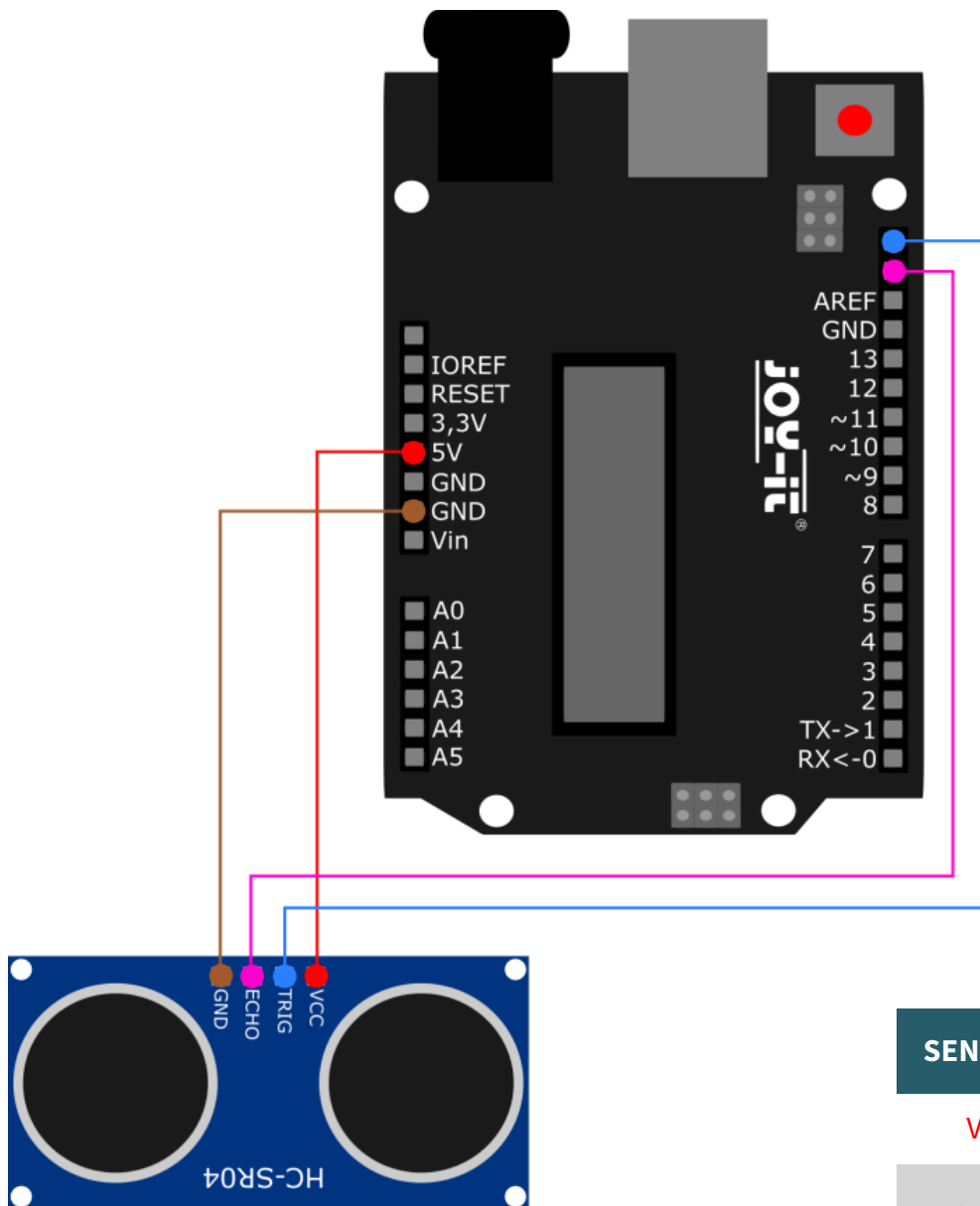
4.1 Anschluss

4.1.1 Schnittstelle GPIO und UART



SEN-US01	Arduino
VCC	3,3 V
Trig / Rx	A5
Echo / Tx	A6
GND	GND

4.1.2 Schnittstelle I2C



SEN-US01	Arduino
VCC	3,3 V
Trig / SCL	D19 (SCL)
Echo / SDA	D18 (SDA)
GND	GND

4.2 Codebeispiele

Wir stellen Ihnen für jede Schnittstelle ein Codebeispiel zur Verfügung. Diese Beispiele können Sie sich [hier](#) herunterladen. Sie können die Codes in Ihrer Arduino IDE öffnen und mittels des **Upload**-Buttons auf Ihren Arduino ziehen. Achten Sie dabei darauf, dass unter **Werkzeuge Board** und **Port** richtig ausgewählt wurden.

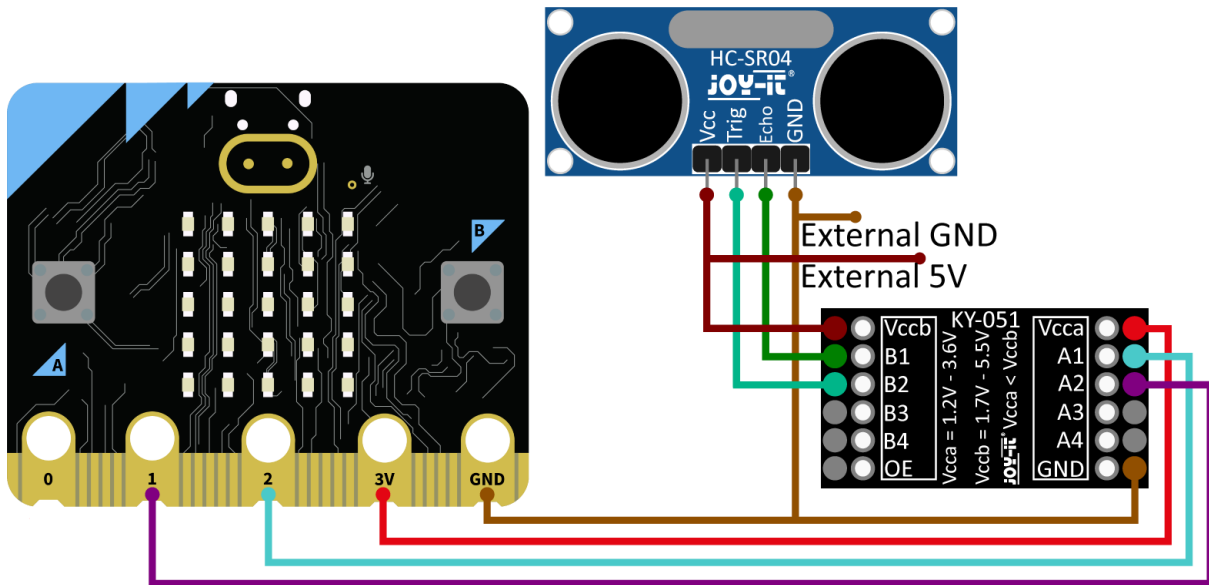


Die serielle Kommunikation zwischen dem SEN-US01 und dem Arduino findet nicht über die Hardware UART-Schnittstelle des Arduinos statt, sondern mittels einer Software-basierenden seriellen Kommunikation. Die UART-Schnittstelle des Arduinos wird nämlich dazu benötigt, die gemessenen Werte auf Ihrem seriellen Monitor anzuzeigen.

5. VERWENDUNG MIT DEM MICRO:BIT

5.1 Anschluss

Schließen Sie Ihren Sensor, wie im Schaubild und der Tabelle zusehen an einen Voltage Translator und Ihren Micro:Bit an. Hierbei empfehlen wir den [KY-051 Voltage Translator von Joy-IT](#).



Micro:Bit	KY-051
3,3 V	Vcca
GND	GND
Pin 2	A1
Pin 1	A2

External	SEN-US01
Extern 5V	+V
Micro:Bit GND + Extern GND	GND

External	KY-051
Extern 5V	Vccb

Sensor	KY-051
Trigger	B1
Echo	B3

5.2 Codebeispiel

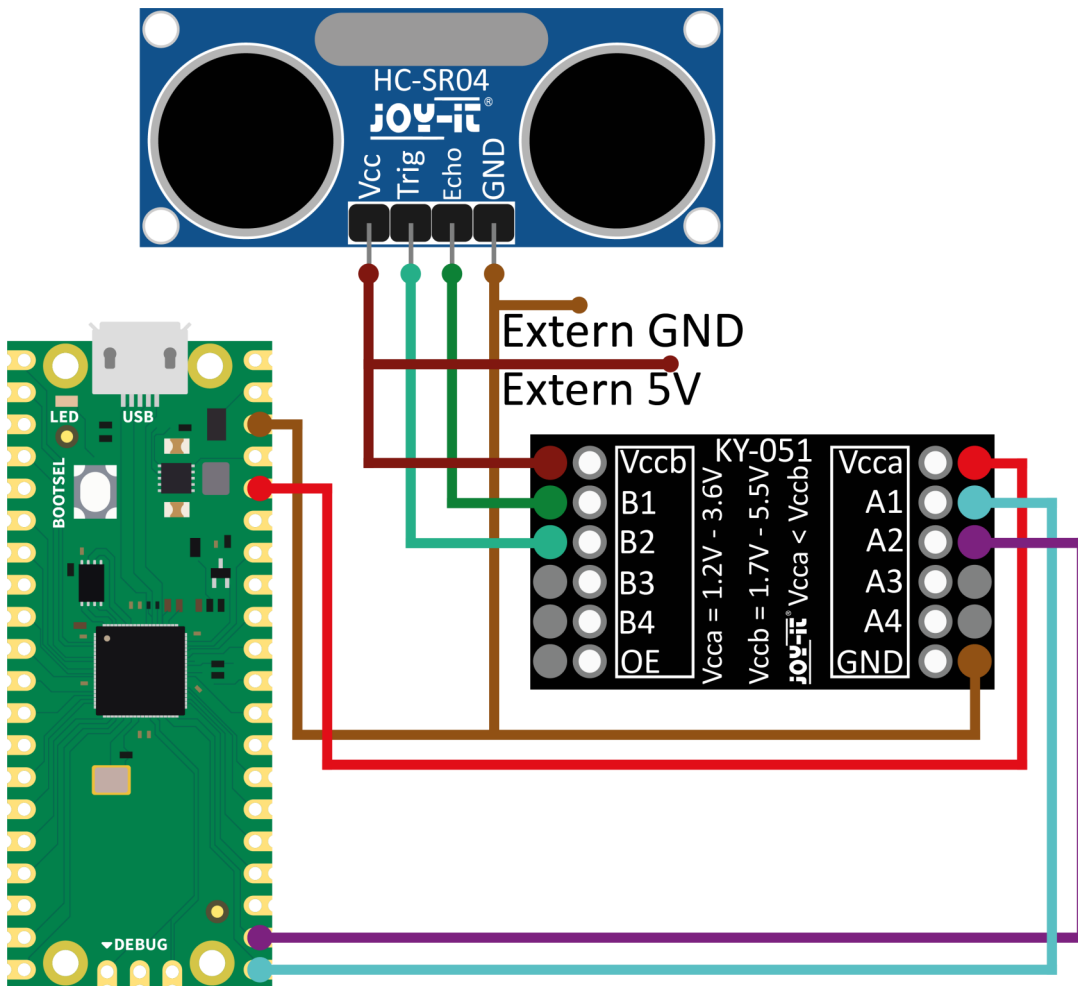
Für das folgende Codebeispiel wird die Erweiterung [pxt-sonar](#) von Microsoft benötigt. Diese können Sie über die Erweiterungen in Ihren MakeCode-Sketch importieren. Das Codebeispiel können Sie dann entweder nachbauen oder über die folgende [.hex-Datei](#) in Ihren Sketch importieren.



6. VERWENDUNG MIT DEM RASPBERRY PI PICO

6.1 Anschluss

Schließen Sie Ihren Sensor, wie im Schaubild und der Tabelle zusehen an einen Voltage Translator und Ihren Raspberry Pi Pico an. Hierbei empfehlen wir den [KY-051 Voltage Translator von Joy-IT](#).



Raspberry Pi Pico	KY-051
5V	Vcca
GND	GND
GPIO 16	A1
GPIO 17	A2

External	SEN-US01
Extern 5V	+V
RPi GND + Extern GND	GND

Sensor	KY-051
Echo	B1
Trigger	B2

External	KY-051
Extern 5V	Vccb

Codebeispiel

In dem nun folgenden Codebeispiel, wird jede Sekunde eine Abstandsmessung durchgeführt und das Ergebnis in der Konsole ausgegeben. Kopieren Sie das Codebeispiel und übertragen Sie es vollständig auf Ihren Raspberry Pi Pico. Alternativ können Sie den Beispielcode auch [hier](#) herunterladen.

```
# Bibliotheken Laden

from machine import Pin
import time

# Initialisierung von GPIO16 als Input und GPIO17 als Ausgang

trig = Pin(17, Pin.OUT)
echo = Pin(16, Pin.IN, Pin.PULL_DOWN)

# Endlosschleife zur Messung der Distanz

while True:

    # Abstandsmessung wird mittels des 10us langen Triggersignals gestartet
    trig.value(0)
    time.sleep(0.1)
    trig.value(1)

    # Nun wird am Echo-Eingang gewartet, bis das Signal aktiviert wurde
    # Danach wird die Zeit gemessen, wie lang es aktiviert bleibt
    time.sleep_us(2)
    trig.value(0)
    while echo.value()==0:
        pulse_start = time.ticks_us()
    while echo.value()==1:
        pulse_end = time.ticks_us()
    pulse_duration = pulse_end - pulse_start

    # Nun wird der Abstand mittels der aufgenommenen Zeit berechnet
    distance = pulse_duration * 17165 / 1000000
    distance = round(distance, 0)

    # Serielle Ausgabe
    print ('Distance:', "{:.0f}".format(distance), 'cm')
    time.sleep(1)
```


7. SONSTIGE INFORMATIONEN

Unsere Informations- und Rücknahmepflichten nach dem Elektrogesetz (ElektroG)

Symbol auf Elektro- und Elektronikgeräten:



Diese durchgestrichene Mülltonne bedeutet, dass Elektro- und Elektronikgeräte **nicht** in den Hausmüll gehören. Sie müssen die Altgeräte an einer Erfassungsstelle abgeben. Vor der Abgabe haben Sie Altbatterien und Altakkumulatoren, die nicht vom Altgerät umschlossen sind, von diesem zu trennen.

Rückgabemöglichkeiten:

Als Endnutzer können Sie beim Kauf eines neuen Gerätes, Ihr Altgerät (das im Wesentlichen die gleiche Funktion wie das bei uns erworbene neue erfüllt) kostenlos zur Entsorgung abgeben. Kleingeräte bei denen keine äußere Abmessungen größer als 25 cm sind können unabhängig vom Kauf eines Neugerätes in haushaltsüblichen Mengen abgeben werden.

Möglichkeit Rückgabe an unserem Firmenstandort während der Öffnungszeiten:

SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

Möglichkeit Rückgabe in Ihrer Nähe:

Wir senden Ihnen eine Paketmarke zu mit der Sie das Gerät kostenlos an uns zurücksenden können. Hierzu wenden Sie sich bitte per E-Mail an Service@joy-it.net oder per Telefon an uns.

Informationen zur Verpackung:

Verpacken Sie Ihr Altgerät bitte transportsicher, sollten Sie kein geeignetes Verpackungsmaterial haben oder kein eigenes nutzen möchten kontaktieren Sie uns, wir lassen Ihnen dann eine geeignete Verpackung zukommen.

8. 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

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

Telefon: +49 (0)2845 9360-50 (Mo - Do: 09:00 - 17:00 Uhr,
Fr: 09:00 - 14:30 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net