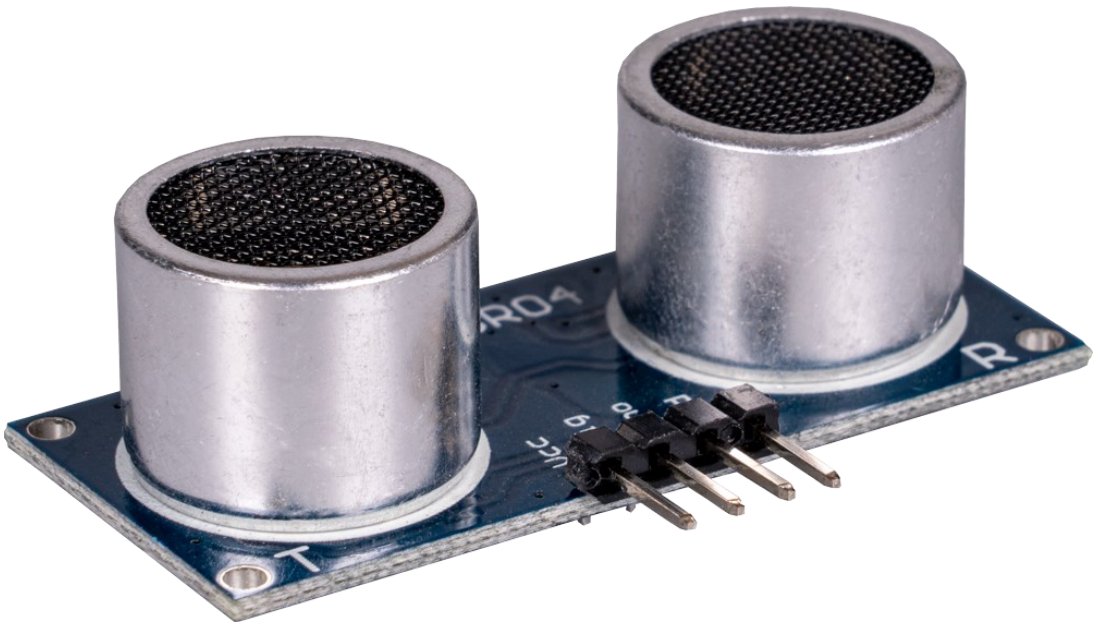
ULTRASONIC DISTANCE SENSOR

SEN-US01



1. GENERAL INFORMATION

Dear customer,

Thank you for purchasing our product. In the following, we will show you which things should be noted during the use.

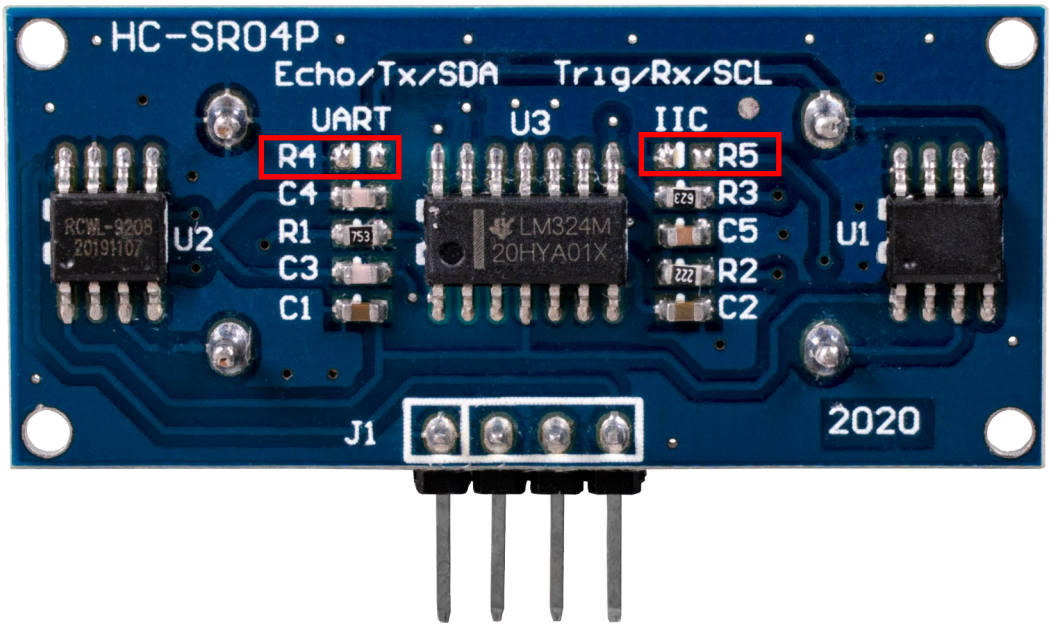
Should you encounter any unexpected problems, do not hesitate to contact us.

This ultrasonic module can measure distances from 2 to 450 cm, with a resolution of 1 mm.

This is done with the help of ultrasound; an ultrasonic signal is emitted that hits an obstacle, is reflected by it and returns to the sensor. The distance of the obstacle can then be calculated from the duration that the signal has been traveling and the speed of sound at which the signal travels.

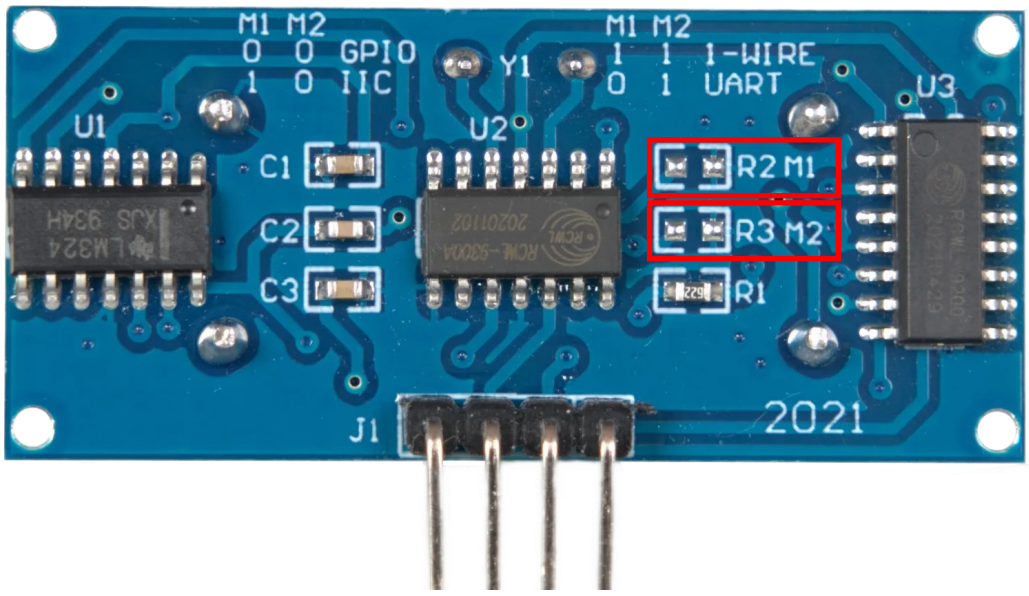
2. INTERFACES

The SEN-US01 has the three interfaces GPIO, UART and I2C. The desired interface can be selected by soldering a resistor. The solder pads R4 and R5 are decisive for this.
If necessary, these must be connected by means of a 10k Ω resistor (0603).



	GPIO	UART	I2C
R4	Not connected	Not connected	10k Ω
R5	Not connected	10k Ω	Not connected

New version with 1-Wire



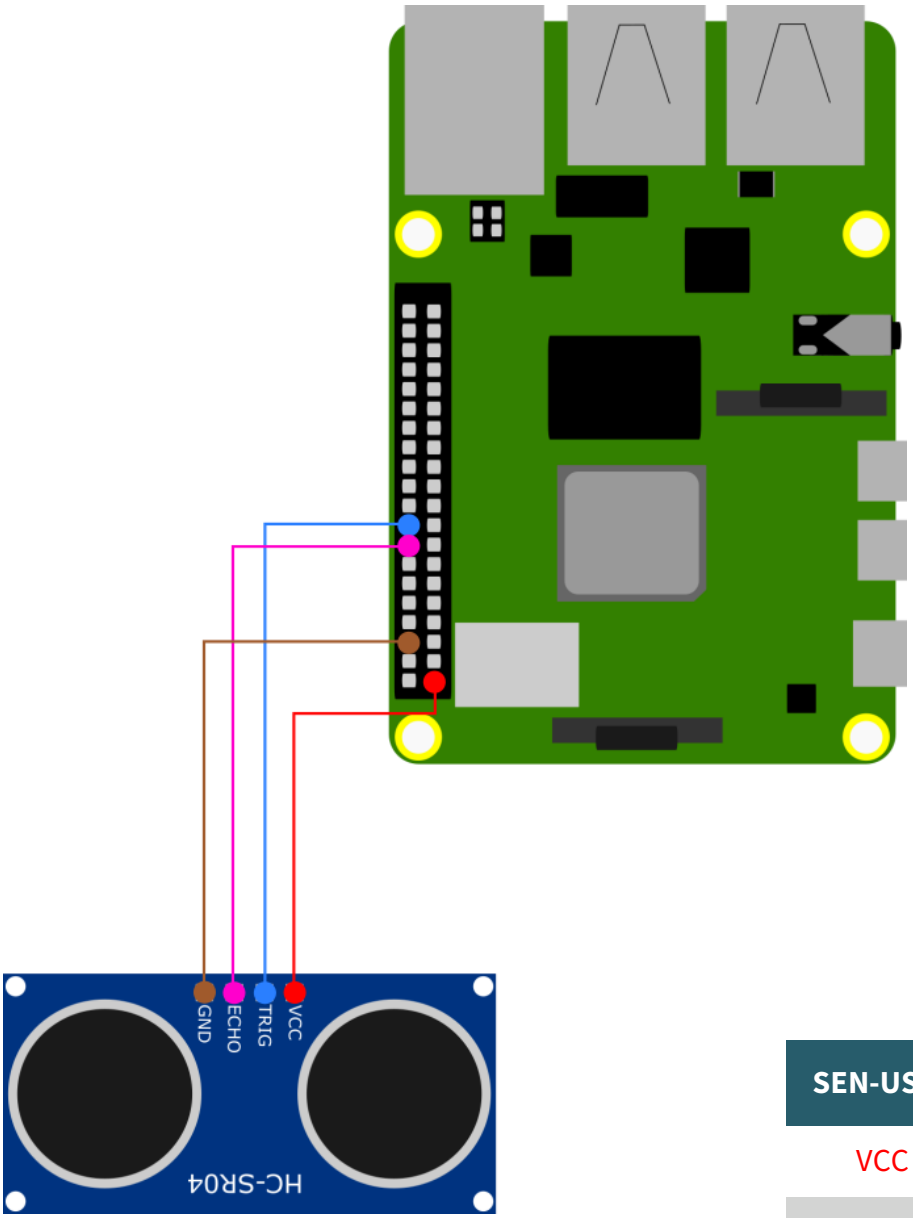
	GPIO	UART	I2C	1-Wire
R2 M1	Not connected	Not connected	10k Ω	10k Ω
R3 M2	Not connected	10k Ω	Not connected	10k Ω

3. USAGE WITH THE RASPBERRY PI

These instructions were written under Raspberry Pi OS Bookworm for the Raspberry Pi 4 and 5. It has not been checked with newer operating systems or hardware.

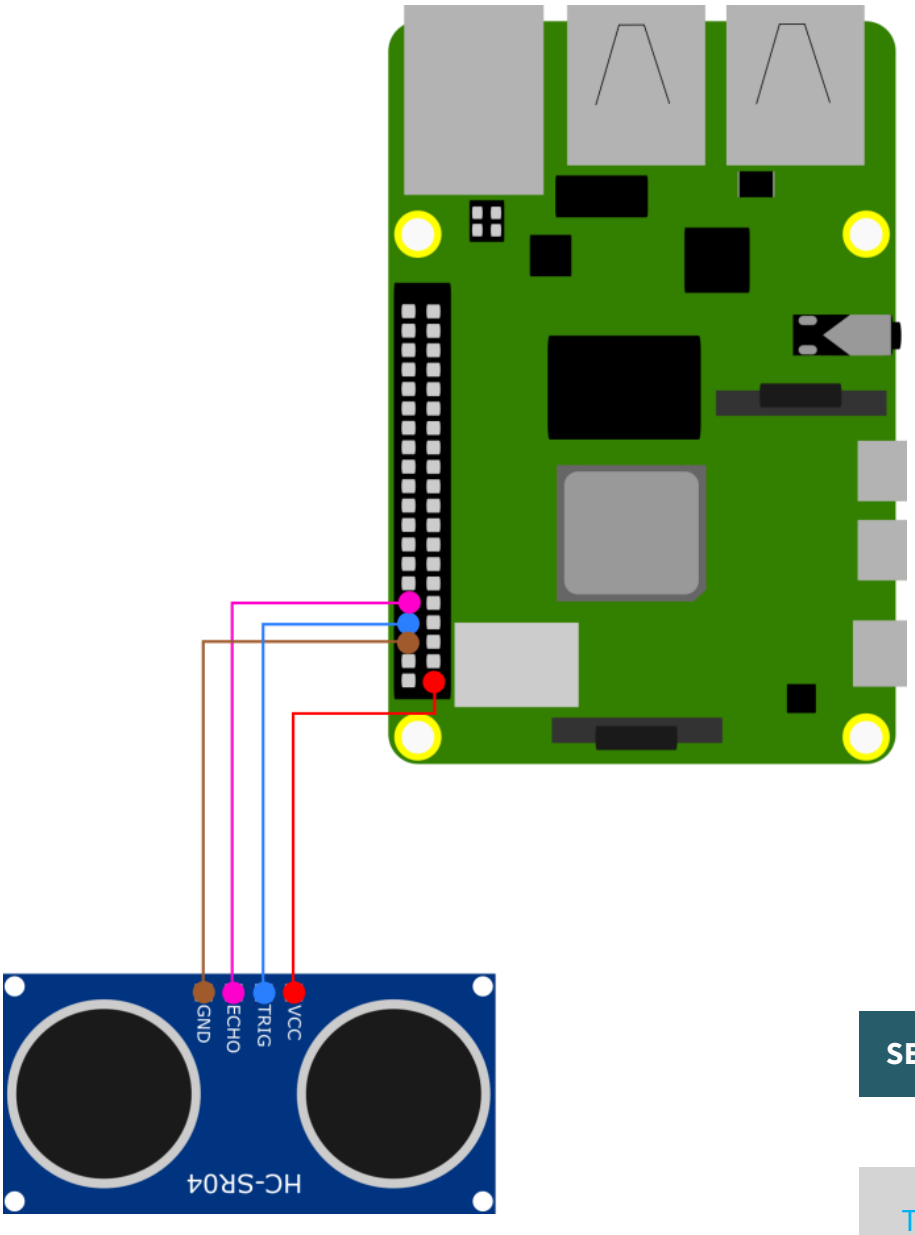
3.1 Connection

3.1.1 GPIO interface



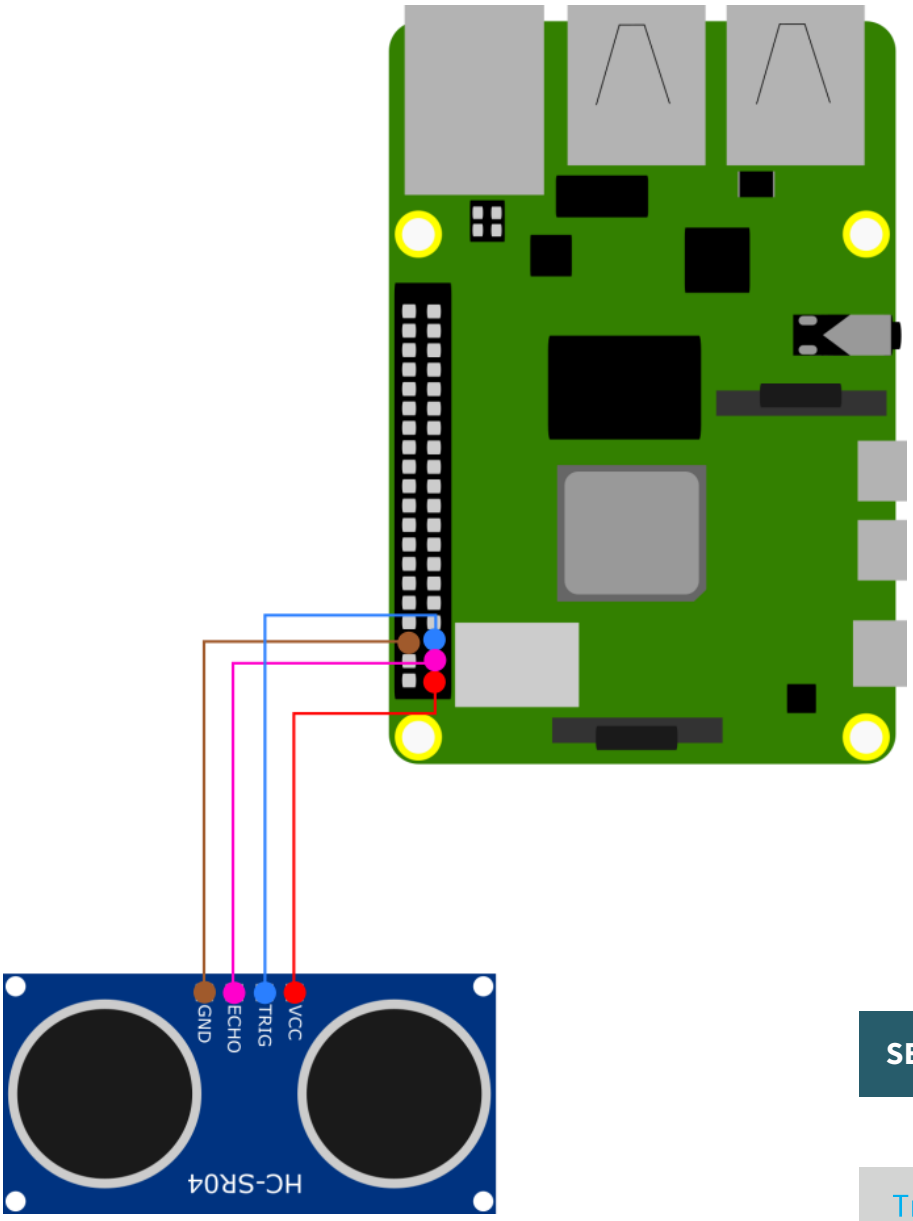
SEN-US01	Raspberry Pi
VCC	3,3 V
Trig	GPIO 24 (Pin 18)
Echo	GPIO 23 (Pin 16)
GND	GND

3.1.2 UART interface



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 I2C interface



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 Code examples

We provide you with a code example which you can download **here** or you can enter the following command into your console.

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

Now unzip the .zip file with the following command:

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

3.2.1 for the GPIO interface

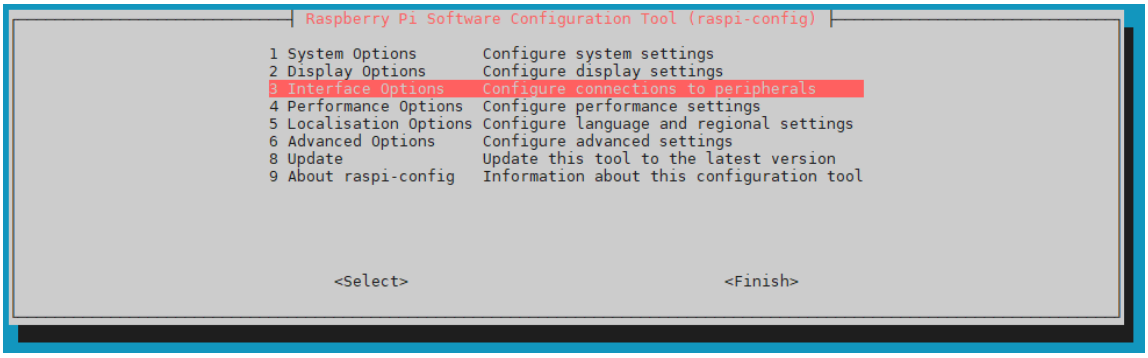
You do not need to perform any further installations for this connection. So you can directly execute the corresponding code example with the following command. Note, that your directory can be different.

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

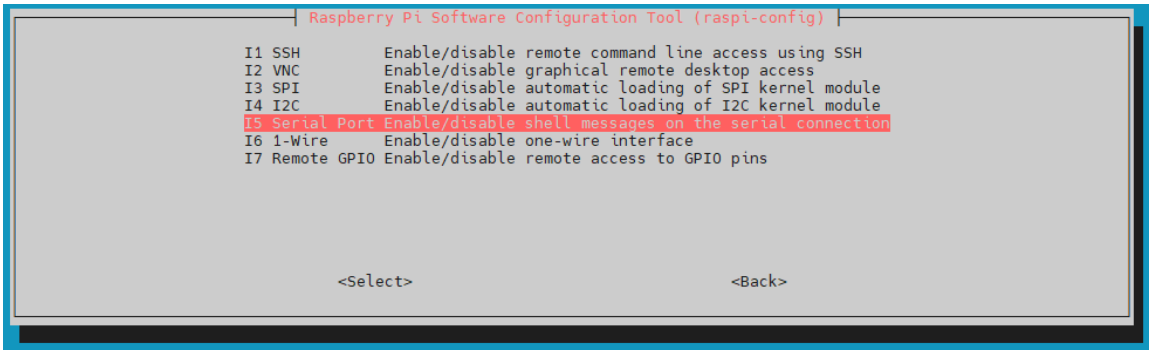
3.2.2 for the UART interface

First you have to enable serial communication in the settings. To do this, enter the following command in the console.

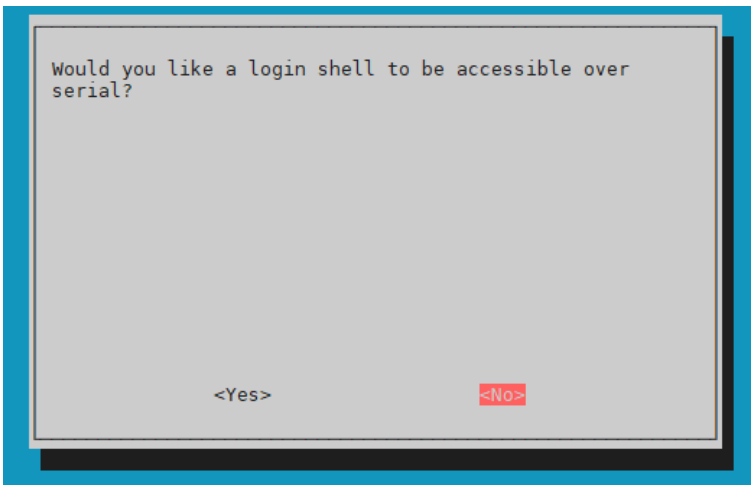
```
sudo raspi-config
```



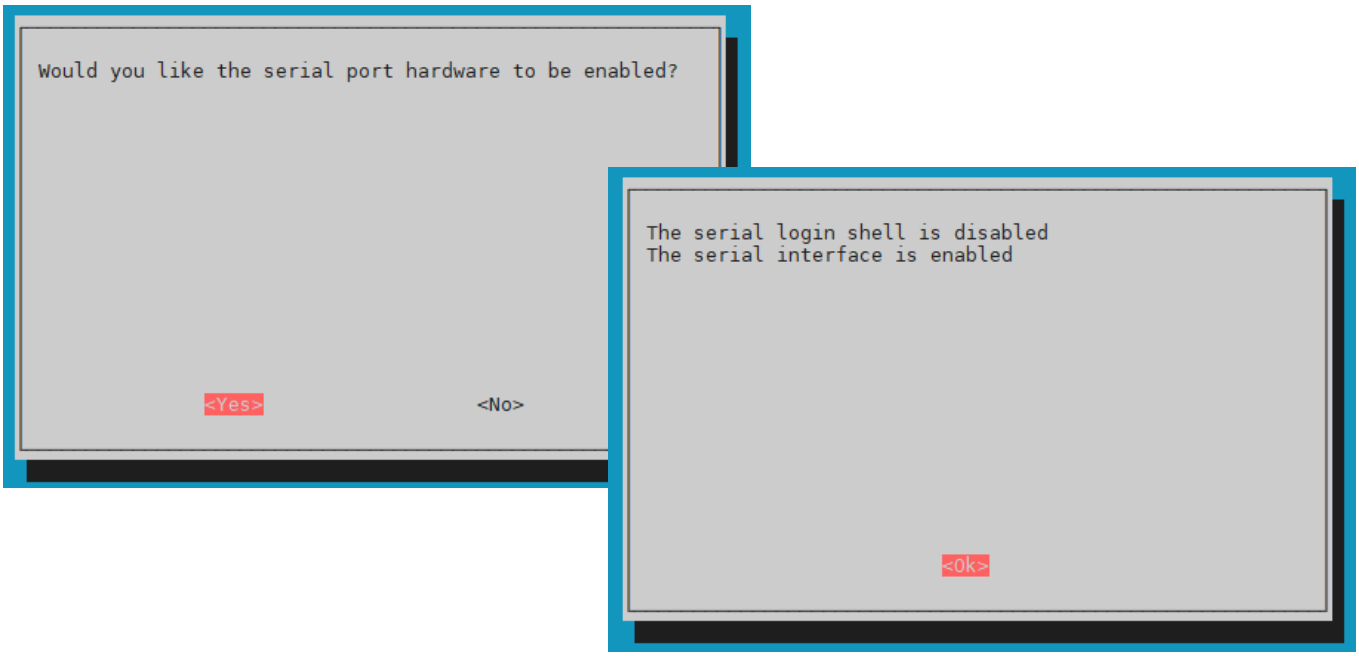
Select there ***Interface Options*** → ***I5 Serial Port***.



There, click **No** when you are asked the question ***Would you like a login shell to be accessible over serial?***.



Accept the question ***Would you like the serial port hardware to be enabled?***.



After that, restart the Raspberry Pi with the following command.

```
sudo reboot
```

Now, execute the following command to install the library for the serial communication.

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

The code example used is configured for the Raspberry Pi 5, to use the Pi4 you need to remove the # character at the beginning of line 6 and insert it at the beginning of line 7.

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

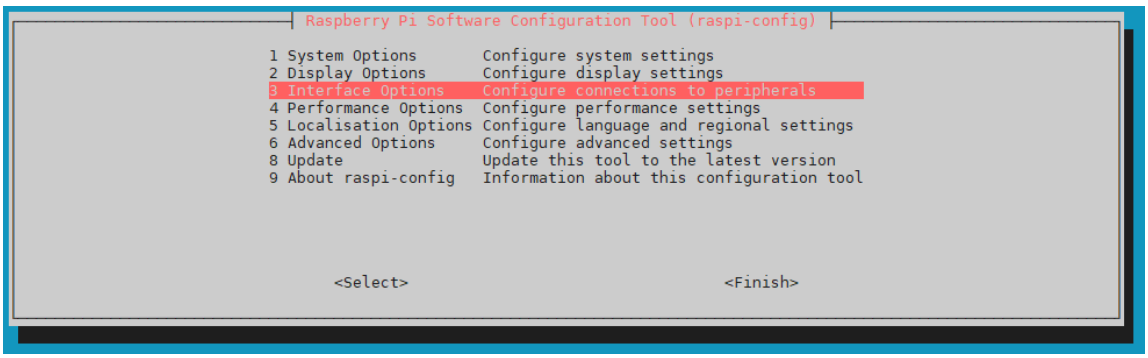
You can now use the following command to run our example code for the UART interface. Be aware that your directory may differ.

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

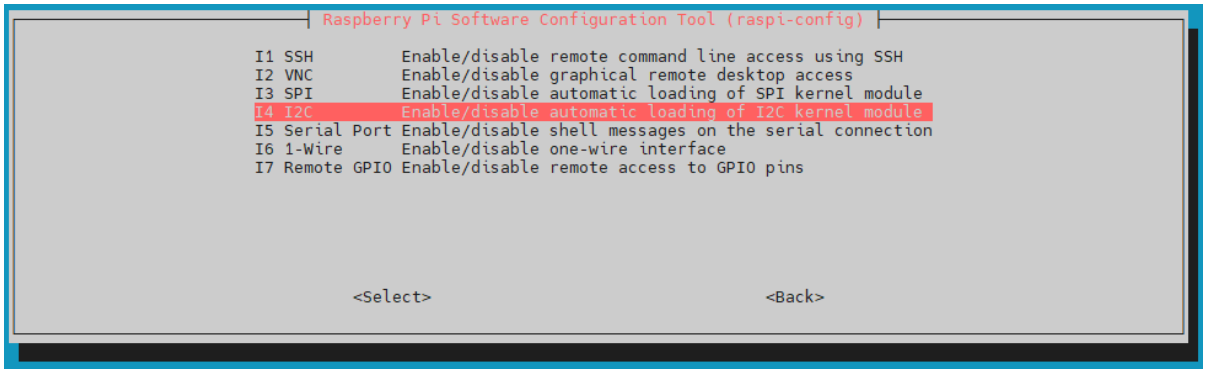
3.2.3 for the I2C interface

First you have to enable serial communication in the settings. To do this, enter the following command in the console.

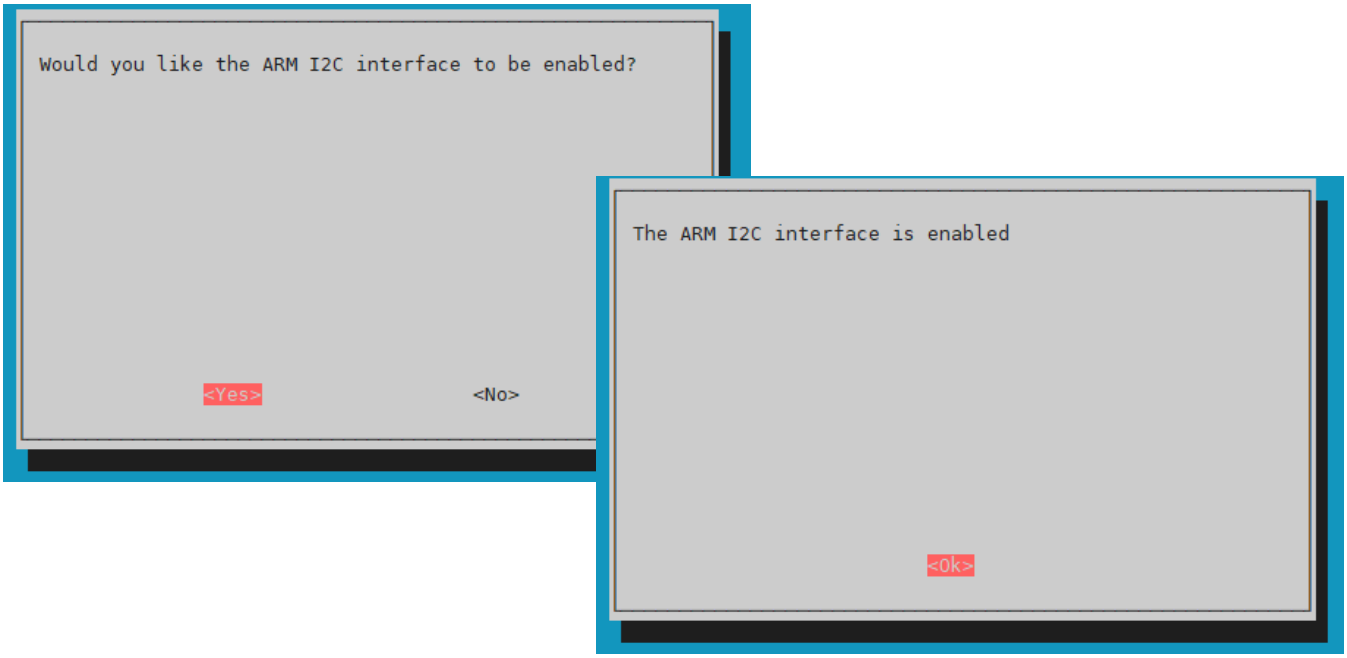
```
sudo raspi-config
```



Select there ***3 Interface Options*** → ***I4 I2C***.



There, click **Yes** when you are asked the question ***Would you like the ARM I2C interface to be enabled?***.



Now, install the required library for I2C communication with the following two commands.

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

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

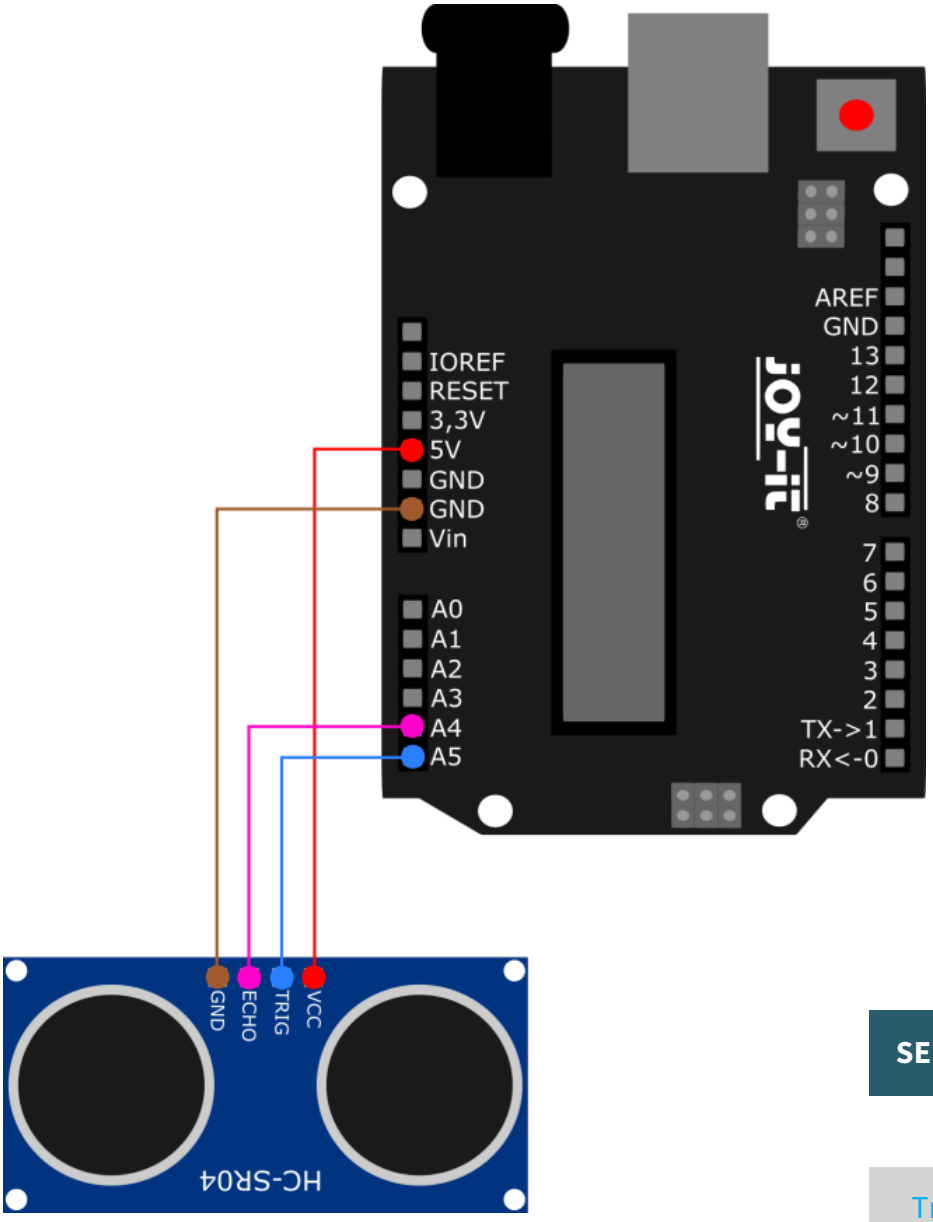
You can now execute our sample code for the I2C interface with the following command. Please note that your directory may differ.

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

4. USAGE WITH THE ARDUINO

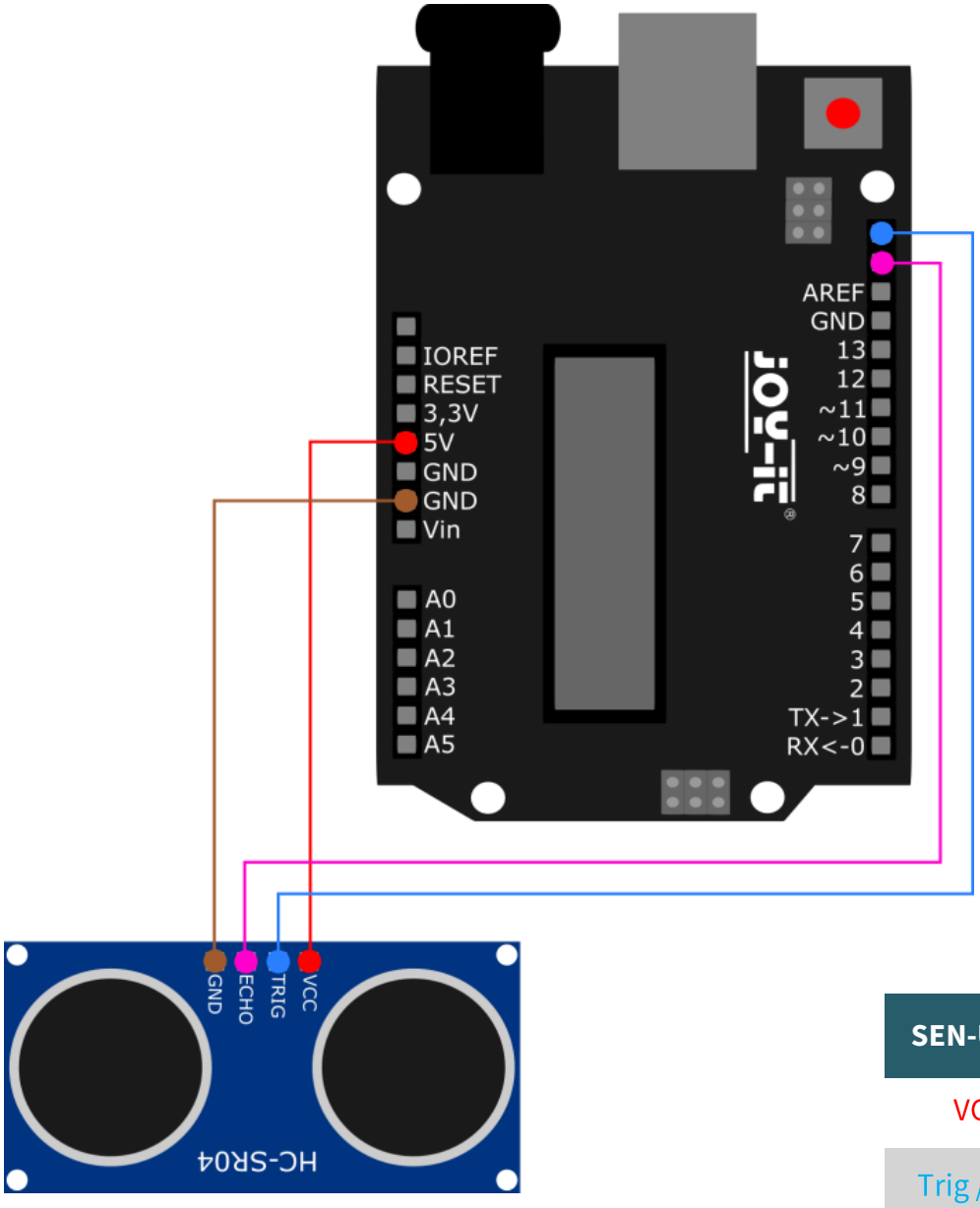
4.1 Connection

4.1.1 GPIO and UART interface



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

4.1.2 I2C interface



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

4.2 Code examples

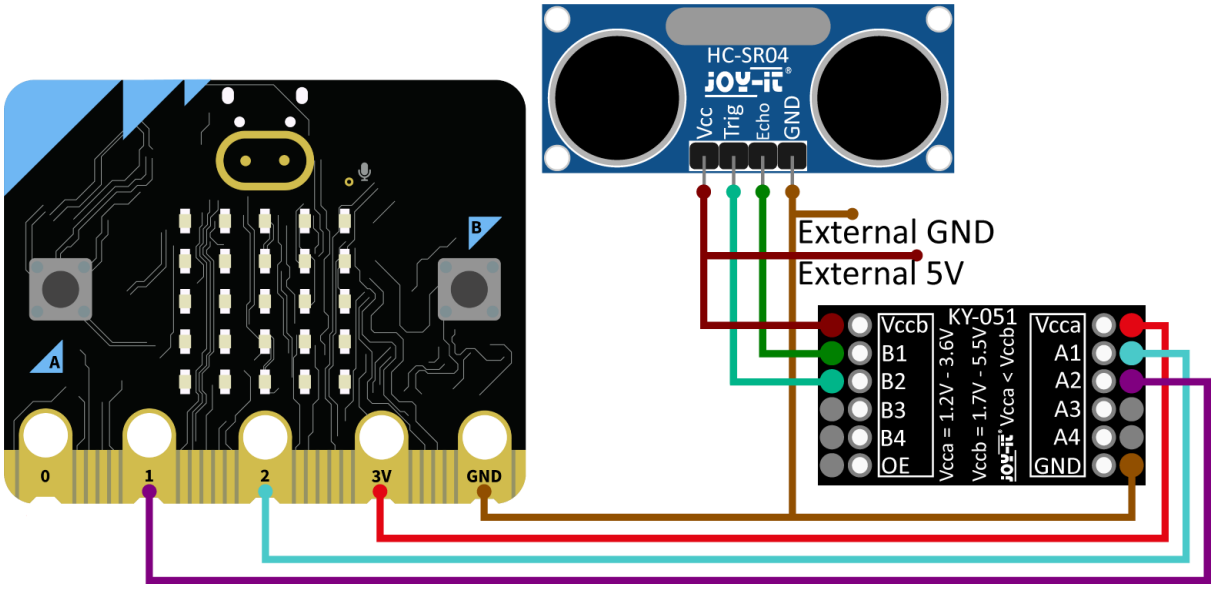
We provide you with a code example for each interface. You can download these examples **here**. You can open these codes in your Arduino IDE and use the **Upload** button to upload them on your Arduino. Make sure that **Board** and **Port** are selected correctly under ***Tools***.

The serial communication between the SEN-US01 and the Arduino does not take place via the hardware UART interface of the Arduino, rather by means of a software-based serial communication. The UART interface of the Arduino is in fact needed to display the measured values on your serial monitor.

5. USAGE WITH THE MICRO:BIT

5.1 Connection

Connect your sensor to a Voltage Translator and your Micro:Bit as shown in the diagram and the table. We recommend the KY-051 Voltage Translator by Joy-IT.



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

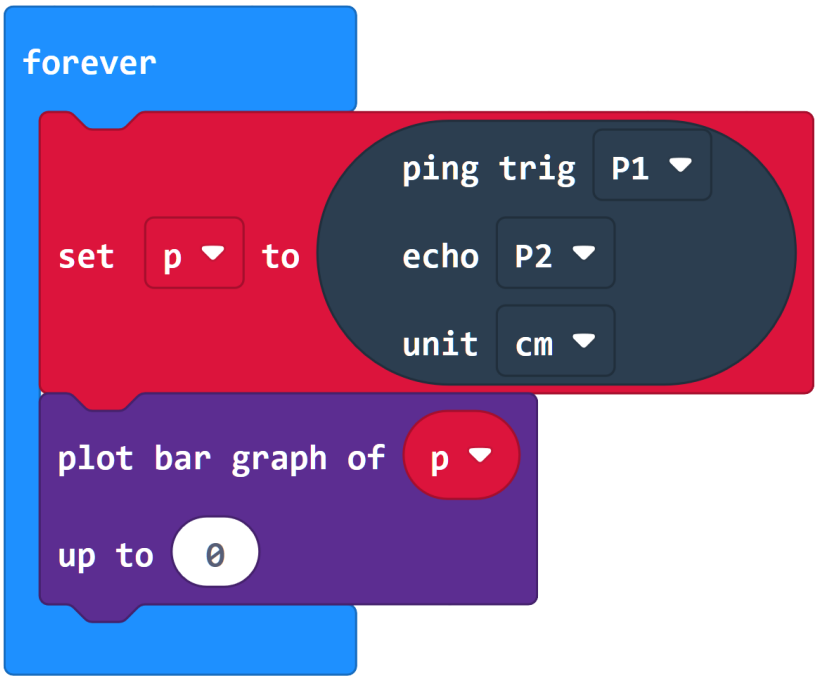
External	SEN-US01
Extern 5V	+V
Micro:Bit GND + Extern GND	GND

External	KY-051
Extern 5V	Vccb

Sensor	KY-051
Trigger	B1
Echo	B3

5.2 Code example

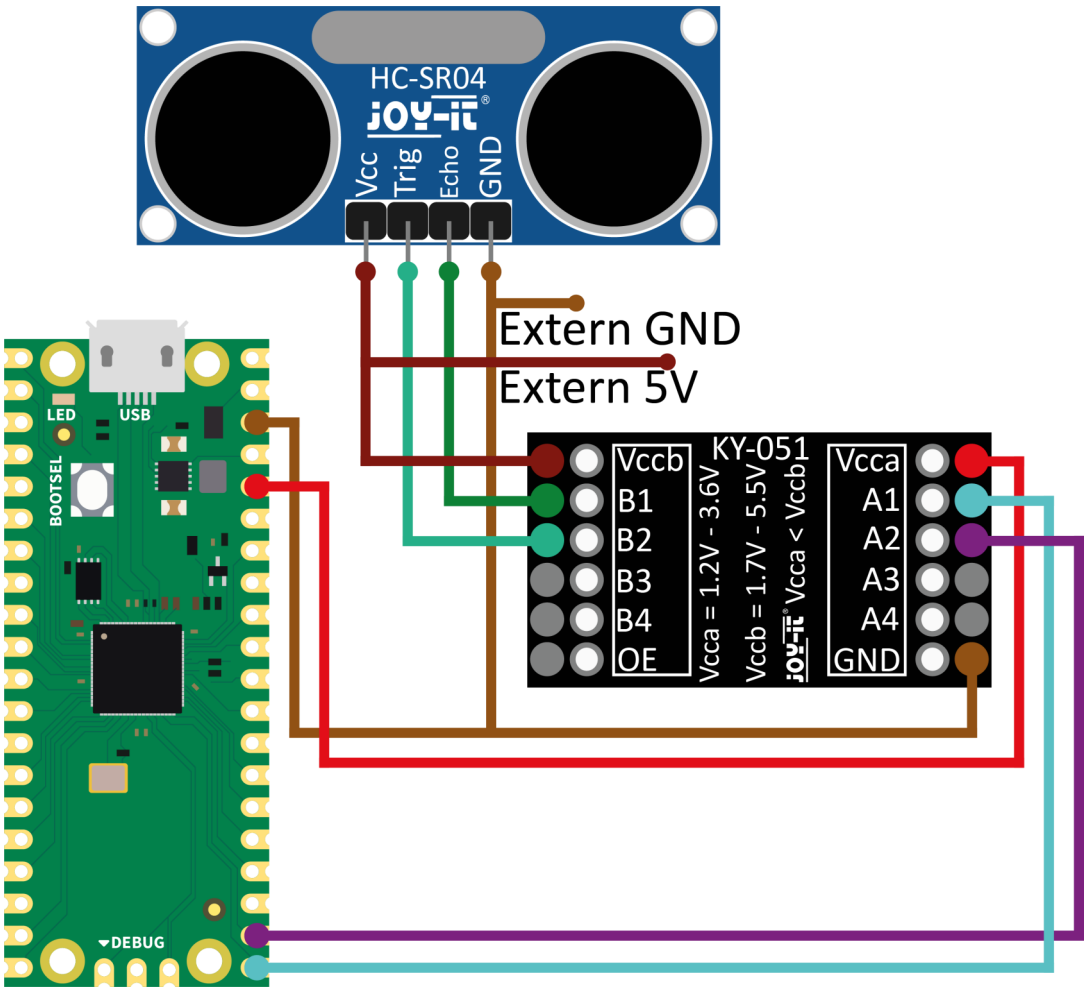
The following code example requires the pxt-sonar extension from Microsoft. You can import this into your MakeCode sketch via the extensions. You can then either rebuild the code example or import it into your sketch via the following .hex-file.



6. USAGE WITH THE RASPBERRY PI PICO

6.1 Connection

Connect your sensor to a Voltage Translator and your Raspberry Pi Pico as shown in the diagram and table. Here we recommend the KY-051 Voltage Translator by 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

External	KY-051
Extern 5V	Vccb

Sensor	KY-051
Echo	B1
Trigger	B2

Code example

In the following code example, a distance measurement is performed every second and the result is displayed in the console. Copy the code example and transfer it completely to your Raspberry Pi Pico. Alternatively, you can download the sample code here.

```
# Load libraries
from machine import Pin
import time

 # Initialization of GPIO16 as input and GPIO17 as output
trig = Pin(17, Pin.OUT)
echo = Pin(16, Pin.IN, Pin.PULL_DOWN)

# Endless loop for measuring the distance
while True:
     # Distance measurement is started by means of the
     # 10us long trigger signal

     trig.value(0)
     time.sleep(0.1)
     trig.value(1)

     # Now waiting at the echo input until the signal has been activated
     # After that, the time is measured how long it remains activated

     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

     # Now the distance is calculated by means of the recorded time
     distance = pulse_duration * 17165 / 1000000
     distance = round(distance, 0)

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

5. OTHER INFORMATION

Our Information and Take-back Obligations according to the Electrical and Electronic Equipment Act (ElektroG)

Symbol on Electrial and Electronic Products:
This crossed-out bin means that electrical and electronic products do not belong into the household waste. You must hand over your old appliance to a registration place. Before you can hand over the old appliance, you must remove used batteries and replacement batteries which are not enclosed by the device.

Return Options:
As the end user, you can hand over your old appliance (which has essentially the same functions as the new one bought with us) free of charge for disposal with the purchase of a new device.
Small devices, which do not have outer dimensions bigger than 25 cm can be handed in for disposal independently of the purchase of a new product in normal household quantities.

1. Possibility of return at our company location during our opening hours
Simac Electronics Handel GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

2. Possibility of return nearby
We will send you a parcel stamp with which you can send us your old appliance free of charge. For this possibility, please contact us via e-mail at service@joy-it.net or via telephone.

Information about Package:
Please package your old appliance safe for transport. Should you not have suitable packaging material or you do not want to use your own material, you can contact us and we will send you an appropriate package.

6. SUPPORT

If any questions remained open or problems may arise after your purchase,we are available by e-mail, telephone and ticket support system to answer these.

E-Mail: service@joy-it.net
Ticket-system: https://support.joy-it.net
Telephone: +49 (0)2845 9360 – 50 (Mo - Do: 09:00 - 17:00 o'clock CET, Fr: 09:00 - 14:30 o'clock CET)

For further information visit our website:
www.joy-it.net

Published: 2024.04.15