**Using a Raspberry Pi distance sensor (ultrasonic sensor HC-SR04)**

For many (outdoor) projects a distance measurement is necessary or advantageous. These small modules are available starting at 1-2 bucks and can measure the distance up to 4-5 meters by ultrasound and are surprisingly accurate. This tutorial shows the connection and control.

## **Hardware**

* HC-SR04 Module ([US](https://www.amazon.com/s/ref=nb_sb_noss_2?tag=754u-20&url=search-alias%3Daps&field-keywords=HC-SR04)\* / [UK](https://www.amazon.co.uk/s/ref=nb_sb_noss/?tag=755-21&url=search-alias%3Daps&field-keywords=HC-SR04)\*)
* Resistors: 330Ω and 470Ω ([US](https://www.amazon.com/s/ref=nb_sb_noss_2?tag=754u-20&url=search-alias%3Daps&field-keywords=restistors+set)\* / [UK](https://www.amazon.co.uk/s/ref=nb_sb_noss/?tag=755-21&url=search-alias%3Daps&field-keywords=restistors+set)\*)
* Jumper wire ([US](https://www.amazon.com/s/ref=nb_sb_noss_2?tag=754u-20&url=search-alias%3Daps&field-keywords=jumper+wire)\* / [UK](https://www.amazon.co.uk/s/ref=nb_sb_noss/?tag=755-21&url=search-alias%3Daps&field-keywords=jumper+wire)\*)

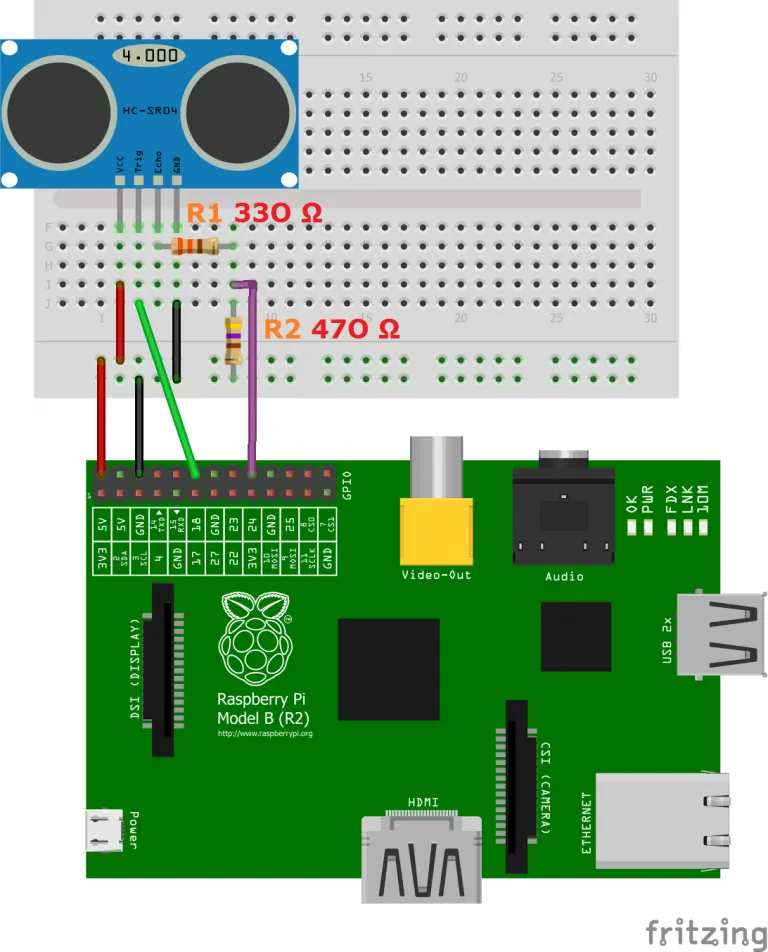
## **Wiring**

There are four pins on the ultrasound module that are connected to the Raspberry:

* VCC to Pin 2 (VCC)
* GND to Pin 6 (GND)
* TRIG to Pin 12 (GPIO18)
* Connect the 330Ω resistor to ECHO.  On its end you connect it to Pin 18 (GPIO24) and through a 470Ω resistor you connect it also to Pin6 (GND).

We do this because the GPIO pins only tolerate maximal 3.3V. The connection to GND is to have a obvious signal on GPIO24. If no pulse is sent, the signal is 0 (through the connection with GND), else it is 1. If there would be no connection to GND, the input would be undefined if no signal is sent (randomly 0 or 1), so ambiguous.

**Here is the structure as a circuit diagram:**



## Script for controlling

First of all, the Python GPIO library should be installed

To use the module, we create a new script

**sudo nano ultrasonic\_distance.py**

With the following content:

#Libraries

import RPi.GPIO as GPIO

import time

#GPIO Mode (BOARD / BCM)

GPIO.setmode(GPIO.BCM)

#set GPIO Pins

GPIO\_TRIGGER = 18

GPIO\_ECHO = 24

#set GPIO direction (IN / OUT)

GPIO.setup(GPIO\_TRIGGER, GPIO.OUT)

GPIO.setup(GPIO\_ECHO, GPIO.IN)

def distance():

    # set Trigger to HIGH

    GPIO.output(GPIO\_TRIGGER, True)

    # set Trigger after 0.01ms to LOW

    time.sleep(0.00001)

    GPIO.output(GPIO\_TRIGGER, False)

    StartTime = time.time()

    StopTime = time.time()

    # save StartTime

    while GPIO.input(GPIO\_ECHO) == 0:

        StartTime = time.time()

    # save time of arrival

    while GPIO.input(GPIO\_ECHO) == 1:

        StopTime = time.time()

    # time difference between start and arrival

    TimeElapsed = StopTime - StartTime

    # multiply with the sonic speed (34300 cm/s)

    # and divide by 2, because there and back

    distance = (TimeElapsed \* 34300) / 2

    return distance

if \_\_name\_\_ == '\_\_main\_\_':

    try:

        while True:

            dist = distance()

            print ("Measured Distance = %.1f cm" % dist)

            time.sleep(1)

        # Reset by pressing CTRL + C

    except KeyboardInterrupt:

        print("Measurement stopped by User")

        GPIO.cleanup()

After that we run:

**sudo python ultrasonic\_distance.py**

**OUTPUT:**

