

hcsr04sensor Python Module

Al Audet

2022-02-17

Contents

1	HC-SR04 Ultrasonic Sensor on Raspberry Pi	3
2	Installation	3
3	Module Description	3
4	Connecting the Sensor	4
5	Accuracy of Readings	5
6	Distance Limitations	6
7	Usage	6
8	Testing the Module	6
8.1	Usage	6
8.2	Sample Output	6
9	Contributing	8
10	Resources	8
11	License	8

1 HC-SR04 Ultrasonic Sensor on Raspberry Pi

<https://www.linuxnorth.org/hcsr04sensor/>

Calculate distance and depth measurements with a HCSR04 Ultrasonic Sound Sensor and a Raspberry Pi. Instructions assume that you are using Raspberry Pi OS/Raspbian Linux.

This module also works with the JSN-SR04T waterproof sound sensor.

2 Installation

Install PIP and RPi-GPIO

```
sudo apt install python3-pip python3-rpi.gpio
```

This will install the apt package RPi.GPIO v0.7.0 from the repository. On Raspberry Pi OS (Bullseye) you may prefer to install the latest version of RPi.GPIO 0.7.1 with Pip. If you prefer the later package run the following;

```
sudo apt remove python3-rpi.gpio <--version 0.7.0  
sudo pip3 install RPi.GPIO <--version 0.7.1
```

Install the hcsr04sensor package. This will also install prerequisite packages if required.

```
sudo pip3 install hcsr04sensor
```

3 Module Description

The module does the following;

- Returns an error corrected distance by using the median reading of a sorted sample. The default sample size is 11 readings.
- You can specify a different sample size by passing `sample_size=x` to `raw_distance`, where x is your desired number of readings. This is useful if you need to lower the sample size to take quicker readings. Beware that the probability of getting erroneous readings increases as sample size is reduced. For my purposes a sample of 11 readings gives a consistent value that I can trust and takes approximately 3 seconds to run with a 0.1 second wait time between individual samples.
- It is also possible to speed up the readings by passing a lower value to `sample_wait` in `raw_distance`.

- The lower the value the quicker the individual samples will be taken. A default of 0.1 is a safe wait time but this can be reduced further. CPU usage increases as faster readings are taken as well as the chance for errors.
- This module uses BCM pin values. See the Raspberry Pi pin layout documentation for your model.
- Uses BCM pin values by default. BOARD pin values are supported.
- Adjusts the reading based on temperature by adjusting the speed of sound.
- Allows measuring distance and depth in metric and imperial units. See - `pydoc hcsrc04sensor.sensor`
- Raises an exception if a faulty cable or sensor prevents an echo pulse from being received.
- Calculate the volume of different types of containers. See recipes for examples.

4 Connecting the Sensor

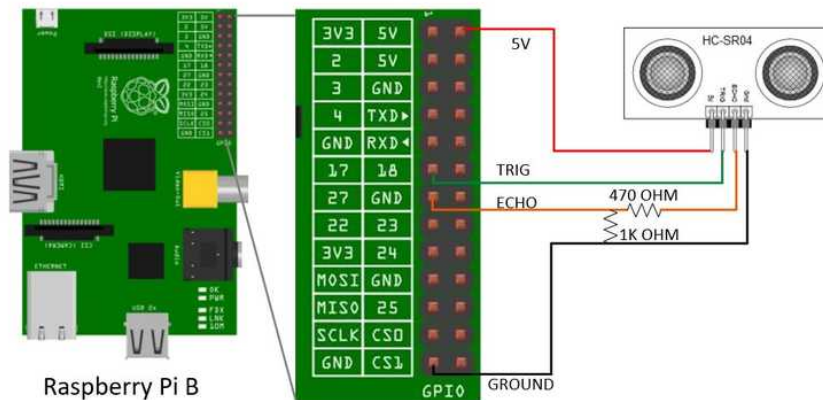
The HC-SR04 sensor has four pins.

- 5V VCC which connects to a 5V pin on the Raspberry Pi
- Trig Pin which connects to a valid GPIO pin. Diagram uses GPIO pin 17.
- Echo Pin which connects to a valid GPIO pin. Diagram uses GPIO pin 27.
- Ground which connects to any valid ground pin on the Raspberry Pi

GPIO pins are rated for 3.3V so you must insert a voltage divider as the power pin on the PI is 5V. In the above diagram a 470 Ohm resistor is soldered on the echo wire. A 1000 Ohm resistor is soldered between the echo and ground wires. This reduces voltage to GPIO pin 27 to 3.4V which is within a tolerable level. Failure to do this can damage your board.

Voltage divider calculator - [courtesy ohmslawcalculator.com](http://courtesyohmslawcalculator.com)

Soldering tutorial -courtesy Xrobots James Bruton



6 Distance Limitations

The HCSR04 sensor is suited for short distance readings. The specification manual says it is suitable up to 13 feet. I have tested it to go further than that, but anything over 12 feet starts having periodic strange readings. This module is not suitable for long distances.

7 Usage

See example scripts in <https://github.com/alaudet/hcsr04sensor/tree/master/recipes>

8 Testing the Module

Run the `/usr/local/bin/hcsr04.py`. This utility does not presently support BOARD pin values. Use BCM pin values.

8.1 Usage

```
usage: hcsr04.py [-h] -t TRIG -e ECHO [-sp SPEED] [-ss SAMPLES]
```

Script tests the HCSR04 sensor under different configurations

optional arguments:

```
-h, --help            show this help message and exit
-t TRIG, --trig TRIG  Trig Pin (Required - must be an integer, must use
BCM pin values)
-e ECHO, --echo ECHO  Echo Pin (Required - must be an integer, must use
BCM pin values)
-sp SPEED, --speed SPEED
Time between individual reading samples (Optional -
must be a float, default is 0.1 seconds)
-ss SAMPLES, --samples SAMPLES
Reading Sample Size (Optional - must be an integer, default is
11)
```

8.2 Sample Output

```
pi@raspberrypi:~ $ hcsr04.py -t 17 -e 27
trig pin = gpio 17
echo pin = gpio 27
```

```
speed = 0.1  
samples = 11
```

```
The imperial distance is 12.3 inches.  
The metric distance is 31.2 centimetres.
```

9 Contributing

The python hcsr04sensor module source code is available on GitHub <https://github.com/alaudet/hcsr04sensor>.

Your contributions are welcome.

1

10 Resources

- Available on GitHub - <https://github.com/alaudet/hcsr04sensor>
- hcsr04sensor Home Page - <https://www.linuxnorth.org/hcsr04sensor/>
- Script recipes - <https://github.com/alaudet/hcsr04sensor/tree/master/recipes>
- HC-SR04 Specification Manual - https://www.linuxnorth.org/raspi-sump/HC-SR04Users_Manual.pdf
- JSN-SR04T Specification Manual - <https://www.linuxnorth.org/raspi-sump/jsn-sr04t.pdf>

11 License

The MIT License (MIT)

Copyright (c) 2022 Al Audet

Permission is hereby granted, free of charge, to any person obtaining a copy of hcsr04sensor and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,

ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR
THE USE OR OTHER DEALINGS IN THE SOFTWARE.