# IC231 Spring 2022 – Lab 2 – Distance Measurement I

Srikanth Sugavanam, Erwin Fuhrer, ssrikanth@iitmandi.ac.in, erwin@iitmandi.ac.in

*In this lab you will learn the fundamentals to use an ultrasound distance measurement module (HC-SR04) to measure the distance of an object using time-of-flight measurement*

## Learning outcomes

In this lab, you will learn how to

- Correctly connect a sensor (HC-SR04) module to the Raspberry Pi
- Implement a hardware-triggered time-based distance measurement.
- How time-of-flight (TOF) measurement works at the example of travel time of sound waves
- Difference of the measurement accuracy between software and hardware-based timer

## Pre-reading

The prereading material enables you to understand the following topics:

- The working principle of the HC-SR04 sensor module
- The pin configuration of the sensor module
- How to interface the sensor module to the Raspberry Pi
- How to drive the Sensor via Python code

Please use the following reading material to familiarise yourself with the topic:

- The working principle, applications and limitations of ultrasonic sensors
  - ➔ https://thepihut.com/blogs/raspberry-pi-tutorials/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi
- User manual of HC-SR04:
  - ➔ https://www.mpja.com/download/hc-sr04_ultrasonic_module_user_guidejohn.pdf
- Speed of sound in air:
  - ➔ https://pages.mtu.edu/~suits/SpeedofSound.html

## Tasks

1. Connect the sensor according to the figure. Why are the resistors required? What are the resistor values that you can use?

2. Place an object 10 cm in front of the sensor.

    a. Connect the Trigger pin to channel 1 and the echo pinto the channel 2 of the oscilloscope.
    b. Send a trigger signal of 10 ms (using RPi.GPIO library) to the sensor.
    c. Adjust the oscilloscope such that you can see the signals on both channels and set the persistence mode. Acquire a single shot signal.
    d. Change the distance of the object to 20 cm and repeat the acquisition.

e. Explain your observations for both channels. For your report save the .csv files and plot the curves and explain how the time-of-flight signal is encoded in your observations.

3. Use the pigpio library to measure the time of flight (Measurement of the Echo Pin) using the Raspberry Pi using the callback functions. Download the code "lab3_3_software".
    a. Download the code and explain the logic. How is the time-of-flight of the sound wave measured? Which pin is used for the measurement? How are the timing events triggerd?
    b. Complete the code by replacing the triple question marks with the correct snippet.
    c. If you have filled the code correctly you receive a numerical output in the shell when you execute the code. Modify the existing code so you receive the output: "There is an object in xx cm distance."

    (**Important**: **For starting the pigpio deamon use only** `sudo pigpiod -s 1`**. Do not set any other parameters as this may freeze your RPi.** )

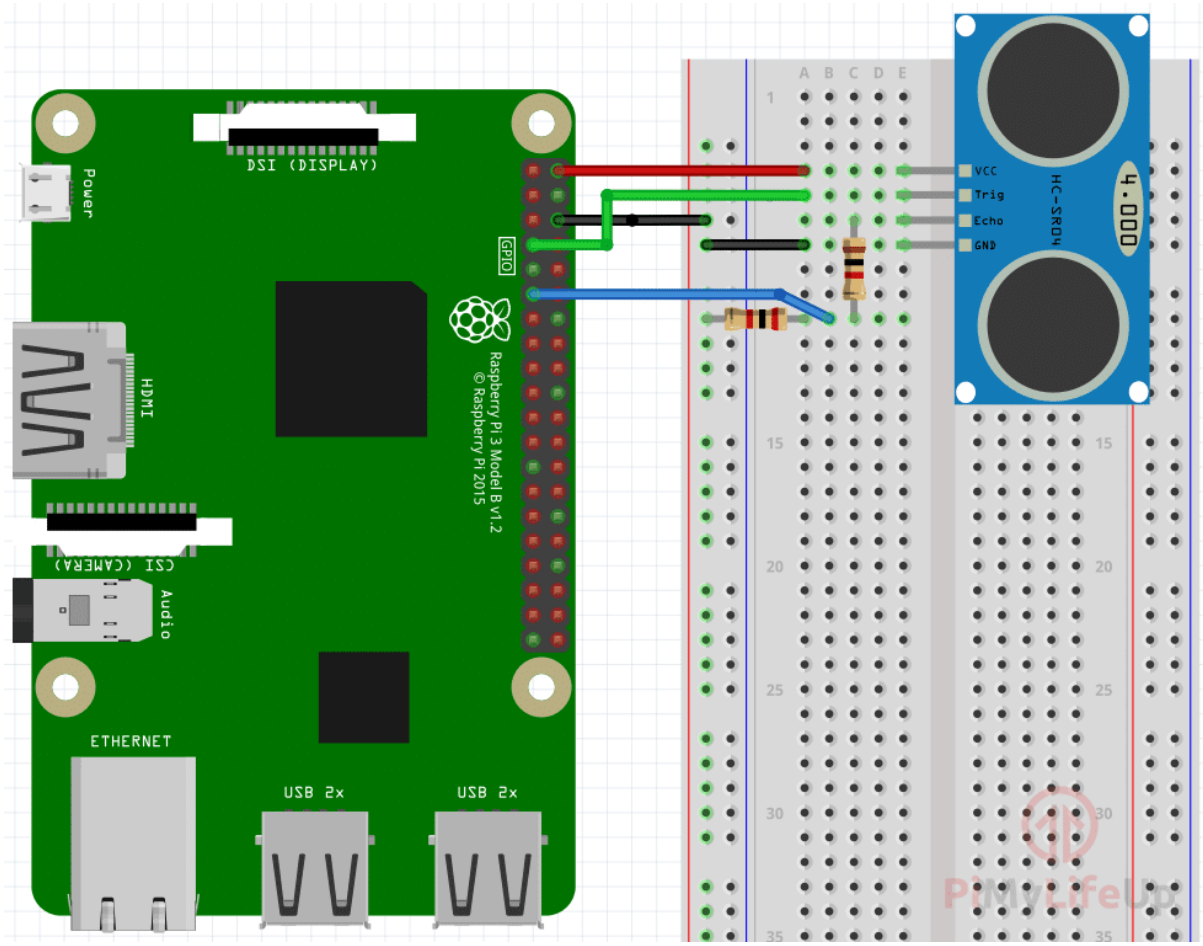
4. The next goal is to compare the accuracy using hardware and software time-based measurements. Place the object at 2 cm, 5 cm and 10 cm and take for each distance 1000 measurement samples. Compute the average and the standard deviation of each distance and compare your results obtained by software-timed code and hardware-timed code. (You may use further libraries of Python numpy or matplotlib)
    a. For the hardware-based measurement you can use the previous code and modify it to automize the acquisition of 1000 measurements.
    b. For the software-based code you can download the template "lab3_4_software" and modify to automize the acquisition of 1000 measurements (Careful: set the TRIG and ECHO pins according to your setup.)

5. Application:
    a. The task is to count the people that **are currently inside a grocery shop**. Assume a distance sensor is placed inside the room on the room ceiling with one meter distance to the entrance. Hence, it can detect people that go through the entrance and measure the distance in a certain range. Create an acquisition scheme that can count the number of incoming and outgoing people and thus measure the total amount of people inside the room.

## Task completion criteria

1. Show proper connection to TAs
2. Show that the displacement of the object leads to change of Echo signal
3. Show the shell ouput of the distance of the object
4. Show the print out of mean and variance of 1000 samples.
5. Show that the counter works for incoming and outgoing objects.

## Circuit connection instructions:




Adapted from: https://pimylifeup.com/raspberry-pi-distance-sensor/

https://pimylifeup.com/raspberry-pi-distance-sensor/

1. Connect the VCC pin of the sensor module to a 5V pin of the Pi.
2. Connect the TRIG pin of the sensor module to a GPIO pin.
3. Connect the ECHO pin of the sensor module to a Pi ground pin by using  by two resistances in series (1k and 2k), hence make a voltage divider.
4. Connect a GPIO pin to the voltage divider (between the 1k and 2k resistor)
5. Connect the GND pin of the sensor module to a ground pin of the Pi.

## Instructions

1. Wait for your TA to signal that the circuit connection is complete.
2. Log on to the Raspberry Pi using VNC Viewer on your computer.
3. Write the program onto the Thonny IDE on the Raspberry Pi.
4. If you run into any issues, ask your TA/Instructor.
5. Recommendations: Generate for each task a new .py-file. In case something goes wrong you can go back to the previous working file.