IC231 Spring 2022 – Lab 3 – Distance Measurement II

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 operate on e

- How to use hardware timer in Raspberry Pi
- How to create microsecond precise PWM in Raspberry Pi
- How to drive the HC-SR044 module using hardware instance ○ How to generate a trigger pulse. ○ How the Transmit mode is triggered ○ How to measure time of flight
- Using HC-SR04 for velocity/acceleration measurement applications

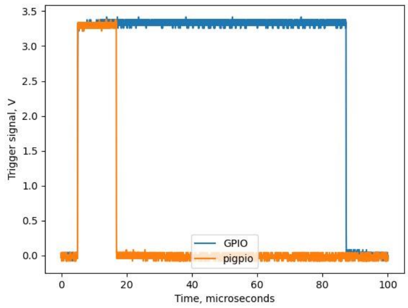
Tasks

1. Download the pigpio **hardware trigger** code from Moodle and run the code directly to observe the output on the oscilloscope.

2. Generate a trigger signal of width 10 μs, first using the RPi.GPIO library. Then, generate a trigger using the gpio trigger function of the pigpio library. You can directly measure the output from a GPIO pin using the oscilloscope probe. It is recommended you use both channels of the oscilloscope to make a simultaneous measurement. In this case you can use the following algorithm to acquire the data.

   Algorithm

   a. Set up Channel 1 of the oscilloscope for a single acquisition.
   b. Use the RPi.GPIO library to generate the trigger signal of 10 microseconds.
   c. Acquire the data from Channel 1, and plot this data (only plt.plot(), don't use plt.show() yet).
   d. Set up Channel 2 of the oscilloscope for a single acquisition.
   e. Use the pigpio code to generate the trigger signal.
   f. Acquire the data from Channel 2, and plot this data. Now use plt.show()



   You should get an output that looks something like the one shown above.

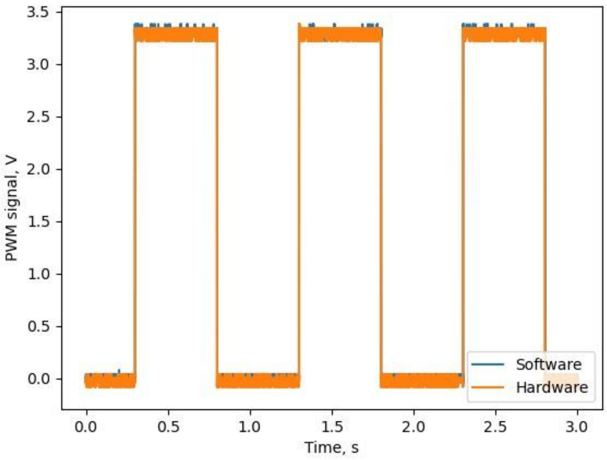
3. Download the pigpio hardware PWM code from Moodle. Run the code directly to observe the output on the oscilloscope.

4. Like in Task 2 above, compare the output of the pigpio-based PWM generation with RPi.GPIO based software PWM generation (i.e. generation of a train of pulses using a for loop). Do this for the following frequencies, all for a 50% duty cycle, showing three periods of the generated PWM signal for each.

   Frequencies to show

   a. 1 Hz
   b. 1 kHz
   c. 10 kHz
   d. 100 kHz
   e. 1 MHz
   f. 10 MHz

   The figure below shows the output for 1 Hz. Retain the same scaled offset and time span for all the frequencies. This will allow for a direct comparison.



5. Calculate the time constant of the gpio pins.
6. Use the pigpio library to generate the trigger for the HC-SR04 sensor, and show the results for three distances.
7. Download the code pigpio_distance_reading.py from Moodle. The code uses two callback functions to measure the time of the rising edge of the echo and the falling edge of the echo. Execute the program and read the time-of-flight. Compare the values that are measured with the RPi with the values obtained from oscilloscope. Explain why you need to wait for the falling edge of the ECHO to calculate the time-of-flight (code line: “#Wait with the code execution….”). Modify the algorithm2 to measure 200 data samples (). How is the sample variance? Compare with the results from Lab 2.

Task completion criteria

1. Show the oscilloscope measurements acquired with RPi
2. Recreate the figure of Task 2 with your own device
3. Show the oscilloscope measurements acquired with RPi
4. Recreate the figure of Task 4 for all requested frequencies
5. Measure the time constant (includes description how you have measured it.).

6. Show the results of time-of-flight measurements
7. Show the acquisition of 200 samples

Pre-reading

- Hardware library in Raspberry Pi: The pigpio library
    - ❾ https://abyz.me.uk/rpi/pigpio/index.html
    - ❾ https://abyz.me.uk/rpi/pigpio/python.html
    - ❾ https://abyz.me.uk/rpi/pigpio/python.html#gpio_trigger
    - ❾ https://clover.coex.tech/en/sonar.html

Circuit connection instructions:

Circuit connection instruction are identical to Lab 2.

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. Generate for each task a new .py-file. In case something goes wrong you can go back to the previous working file.

Challenge exercise

Find the maximum time span of acquisition of the oscilloscope, i.e. the maximum time from starting the trigger signal to finishing the echo. Knowing this value, and the measurement parameters of the HC-SR04, generate a PWM pulse train using the pigpio library to continuously measure echoes over this time span of acquisition of the oscilloscope. Acquire both the trigger and echo pulses using the oscilloscope (***you don't have to measure the echo using the Raspberry Pi GPIO***).

Answer the following questions

a. Why do you want to generate a train of PWM pulses using the pigpio library?
b. What is the maximum number of echo/trigger pulses you can measure using this approach?
c. How can you calculate the distance from the echo signals?
d. What are the problems and challenges you see in this form of measurement? How can these problems be mitigated?
e. Using the above configuration, formulate a method for measuring the velocity and acceleration of an object moving towards and away from the sensor.