Experiment 3: Measuring Speed Using the HC-SR04

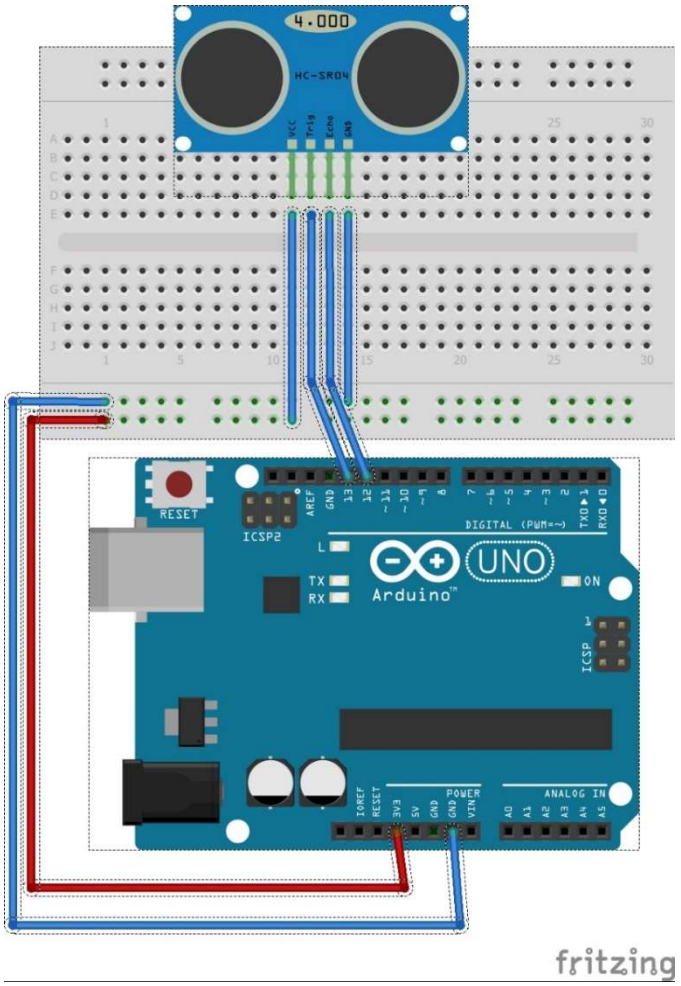
In this experiment we will learn one very good way to measure distance and speed. This will involve the popular (and inexpensive) HC-SR04 sonic ranger that is included with your Arduino kit. We'll learn how it works, how to connect it, and how to program it for distance and speed measurements. We'll also see a few new coding and uncertainty analysis "tricks" along the way.

Learning Objectives: In order to earn high marks on this experiment, the student should be able to:

- Connect the HC-SR04 sonic ranger to the Arduino using a breadboard and wires.
- Program the HC-SR04 sonic ranger to measure echo duration and distance.
- Apply rules of error/uncertainty propagation needed for data analysis.
- Determine, by measurement, calculation and graphical analysis, the speed of sound.
- Plot data on a graph and analyze its slope.
- Compute the uncertainty of the slope of a line using LINEST.

Part A: The HC-SR04 Sonic Ranger

The HC-SR04 ultrasonic sensor is widely available and costs less than a Happy Meal at McDonald's[1]. They remind me of a bat[2] - one side emits a high frequency sound wave and the other side detects the echo after it bounces back off the object we are sensing. The device measures the amount of time[3] it takes for the sound chirp[4] to go out, reflect, and return. The setup is pretty straightforward - there are four connections you'll have to make as shown in the diagram.



Our objective is to bounce the sound off of a hard surface and measure the time it takes for the echo to return to the sensor. If we know that amount of time and the distance from the sensor to the hard surface, we can determine the speed of sound!

[1] I got five of them on Amazon Prime for $9.79.
[2] The kind that are furry and fly and hang upside down at night. Not the kind that are used to hit baseballs.
[3] In microseconds. 1s = 1x10^6 μs.
[4] Since the sound chirp is at about 40kHz, *you* can't hear it…humans can typically only hear frequencies up to about 20kHz.

1. After making the proper wiring connections, open a brand new Arduino sketch. Coding this little gem requires some of our usual tricks (and a few new ones)…

- Pin 13 (connected to `Trigger`) is a digital `OUTPUT`.
- Pin 12 (connected to `Echo`) is a digital `INPUT`.
- So in the `void setup()` you'll have to make that happen.
- Then, in the `void loop()` you'll need to trigger the thing to send out its chirp. You do this by first setting the `Trigger` pin `LOW`, then setting it `HIGH` for 10 microseconds, then setting it `LOW` again.
- Then get the pulse duration (which is the Elapsed Time we needed above) by taking input on the `Echo` pin (on pin 12). The command for this is `pulseIn(echoPin, HIGH).`
- The "code" below is "pseudocode" meaning that you'll need to think about what command to use for each line. Some potentially useful command lines can be found in the text box.

Potentially useful lines of code

```
digitalWrite(13, LOW);
pinMode(13, OUTPUT);
Serial.begin(9600);
pinMode(12, INPUT);
digitalWrite(13,HIGH);
delayMicroseconds(10);
delay(3000);
Serial.println("#goPhysics");
```

```
void setup()
{
    Put a line of code to set serial communication at 9600 baud
    Put a line of code here to make pin 13 an output
    Put a line of code here to make pin 12 an input
}
void loop()
{
    Put a line of code here to set the Trigger pin LOW
    delayMicroseconds(2);
    Put a line of code here to set the Trigger pin HIGH
    Put a line of code here to wait/delay for 10 microseconds
    Put a line of code here to set the Trigger pin back to LOW again

    long duration = pulseIn(12, HIGH);
    Serial.print(duration);
    Serial.print(" ");
    Serial.println("microseconds");
    Put a line of code here to delay for a few seconds
}
```

2. Place a reflecting surface a meter or two from the sensor. Measure the distance from the sensor to the surface making sure to approximate the uncertainty in this distance.
3. Record a set of "echo times" from the Serial monitor of your Arduino/`HC-SR04` setup. Compute this time and its associated uncertainty.
4. Use your measurements to determine the speed of sound in air[5].

[5] Don't forget that the sound had to travel to the reflecting surface and back…a "round trip"!

Part B. Determining the Uncertainty and Rules for Propagation of Error

Now that you have determined the speed of sound you're wondering just how precise your result is? How much uncertainty is there is your answer? Well, if you think about it, the uncertainty in your answer is the result of the uncertainties of the two quantities (namely echo time and distance) you used to compute the speed of sound. But how do these individual errors propagate to the error of the answer?

Well, in this course we will use a set of rules for error propagation. There are two main rules you'll need to apply….

- **Addition or Subtraction:** If $Z = X \pm Y$ you add the absolute uncertainties so we have…

$$\delta Z = \delta X + \delta Y$$

- **Multiplication or Division:** If $Z = XY$ or $Z = X/Y$ you add the percent uncertainties so we have…

$$\frac{\delta Z}{Z} = \frac{\delta X}{X} + \frac{\delta Y}{Y}$$

Example: A person measures their heart rate by counting $B = 40 \pm 1$ beats in $T = 30.0 \pm 0.5$s. Determine the heart rate and uncertainty.

  - We have $\frac{\delta B}{B} = \frac{1}{40} = 0.025 = 2.5\%$ and $\frac{\delta T}{T} = \frac{0.5s}{30.0s} = 0.017 = 1.7\%$ so we know that the percent uncertainty of the heart rate measurement is 2.5% + 1.7% = 4.2%.
  - The heart rate is $R = B/T$ = 40 beats/0.5 minutes = 80 Beats/Minute = 80 BPM.
  - The absolute uncertainty is δR = 80BPM * 4.2% = 80BPM * 0.042 = 3 BPM
  - The final reported result is, therefore $R = 80 \pm 3$ BPM.

1. Determine the percent and absolute uncertainty of your speed of sound measurement.
2. Report your speed of sound with its associated absolute uncertainty!

Part C: Another Way

In parts A and B of the lab, you determined the speed of sound using a single distance and echo time. Another, perhaps superior method, would be to collect echo times for a range of distances, plot distance versus echo time and use a graphical method to determine the speed of sound.

1. Position your reflecting surface at a range of distances from the sensor. Measure the distance and echo time at each of these distances.
2. Now make a plot of position versus echo time using Excel.
3. Use the Excel `LINEST` function to determine the slope of the trendline (as well as its uncertainty).
4. Does your result "agree" with your result from part B? How do you know?