
Aberystwyth Robotics Club - Arduino Tutorials - Ultrasonic Tutorial

Introduction

This example shows you how to find the distance from an obstacle using ultrasonic sensors, for now we will return the values from the ultrasonic to the serial monitor on the Arduino.

Hardware Required

- Arduino or Genuino Board
- HC-SR04 Ultrasonic
- Jumper Cables
- Optional breadboard

Circuit

Ultrasonics work by sending out pulses of sound and then counting how long it takes for those pulses to come back. So an Ultrasonic sensor actually has two parts: a transmitter and a receiver.

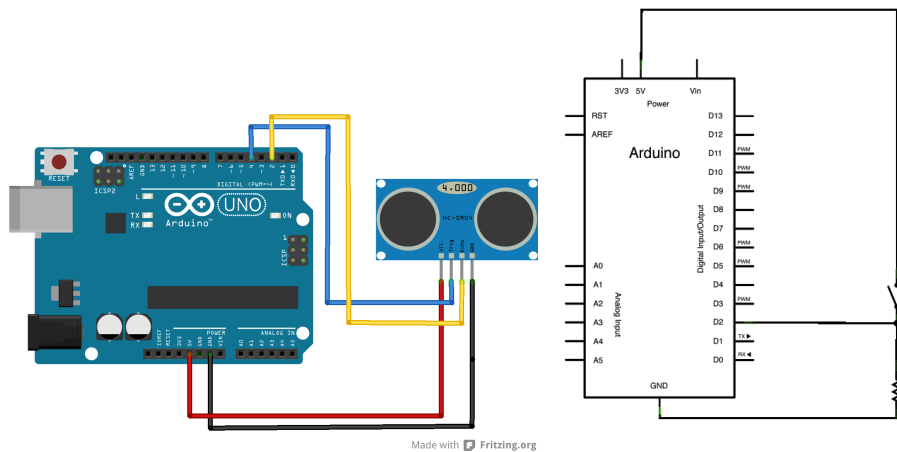
If you look at the sensor it has two round bits, labelled T for transmitter and R for receiver. (In electronics things which both transmit and receive signals are often called "Transceivers" so this is really an Ultrasonic transceiver.)

The Ultrasonic sensor has 4 wires labelled Vcc, Trig, Echo and Gnd.

These are for the voltage, the trigger of the sonar, the echo detection signal, and the ground.

1. **Wire up the the first wire (voltage, or Vcc) to the 5v output of your Arduino.**
2. **Wire up the trigger or Trig to data pin 2 on your Arduino.**
3. **Wire up the echo detection signal to data pin 3 on your Arduino.**





Code

Remember that all Arduino programs have two parts: the setup which happens once, and is used to set up anything you need to use in the program. This is where you tell the Arduino which components are wired up to which pin, and stuff like that. There is also a loop, and the loop is repeated again and again until the Arduino is turned off (or runs out of battery power).

In the setup function here we need to tell the Arduino which pin has the Trig connection, and which pin has the Echo connection.

We are going to use the trigger pin to send a message to the sensor from the Arduino (so that is an OUTPUT pin), but we are going to use the Echo pin to read a message from the sensor (so that is an INPUT pin). Code from <https://create.arduino.cc/projecthub/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-036380>

1. Plug in your Arduino or Genuino board.
2. Start the Arduino Software (IDE).
3. Enter the code below in the setup function.

```
Serial.begin(9600);
pinMode(2, OUTPUT);
pinMode(3, INPUT);
```

As well as telling the program what's connected to what pin, it also sets up the serial monitor so we are ready to start monitoring stuff.

4. Enter the following code into the loop function.

```
long duration;
digitalWrite(2, HIGH);
delayMicroseconds(10);
digitalWrite(2, LOW);
duration = pulseIn(3, HIGH);
Serial.print(duration);
Serial.println("_:_duration_");
```

This triggers the sensor to send out a signal, then reads the signal coming back from the sensor.

There is a lot going on in this program.



- First we setup a variable called duration to hold the length of time between the sound going out and the sound bouncing back.
- Then we trigger a quick pulse of sound (by writing HIGH to the trigger pin) - we only do this for a tiny amount of time (10 microseconds) before turning it off again by writing LOW to the trigger pin.
- We then capture the time from the pulse being sent out to coming back by looking for a HIGH pulseIn on the echo pin.
- Finally we print the duration out to the serial monitor.

5. **Test the program by clicking the tick.**

6. **Put the program on the Arduino using the arrow button.**

7. **Open the serial monitor.**

To open the serial monitor, click on the button that looks a bit like a magnifying glass in the top right corner of the Arduino IDE window (it will say "Serial Monitor" when you hover over it). That will open up a new window which contains the stuff you have printed to Serial. You should see something like this:

```
386: duration
417: duration
381: duration
```

These are measurements of how long it took (in microseconds) for the sound to bounce off a thing in the world and then back to the ultrasonic sensor.

Going from a measurement of time to a measurement of distance

1. **Set up a barrier near your computer, and get hold of a ruler.**
2. **Measure the distance from the sensor to the barrier, and fill in a table of measurements from the serial monitor.**
 - (a) If the barrier is 10cm from the sensor, how long does it take the sound to travel?
 - (b) Take some more measurements. Can you come up with a way of converting "sound-travel-time" into centimetres?
 - (c) If you can come up with a useful conversion system, can you then get that into your program - that is, can you work out how to get your program to give an output in centimetres? Give it a go, and then test your ultrasonic ruler against a real ruler.

