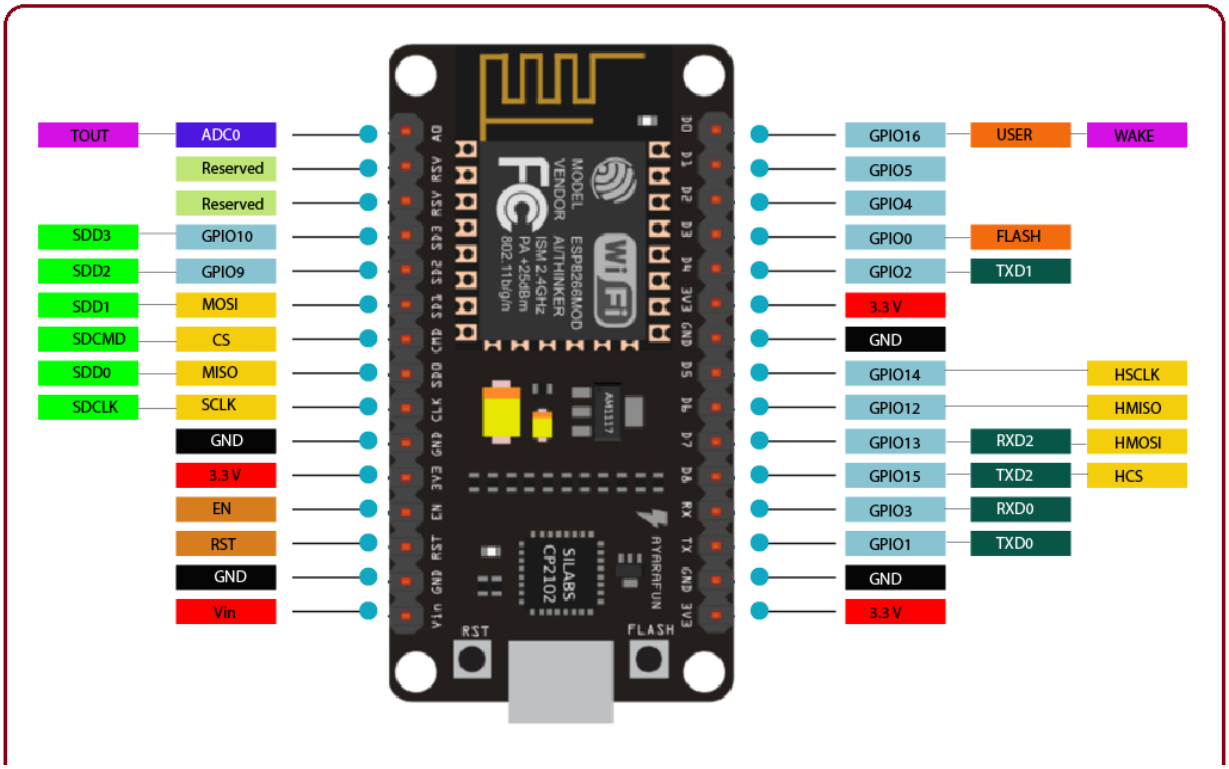
# ESP8266 – Tutorial Demonstration III

## You will need:

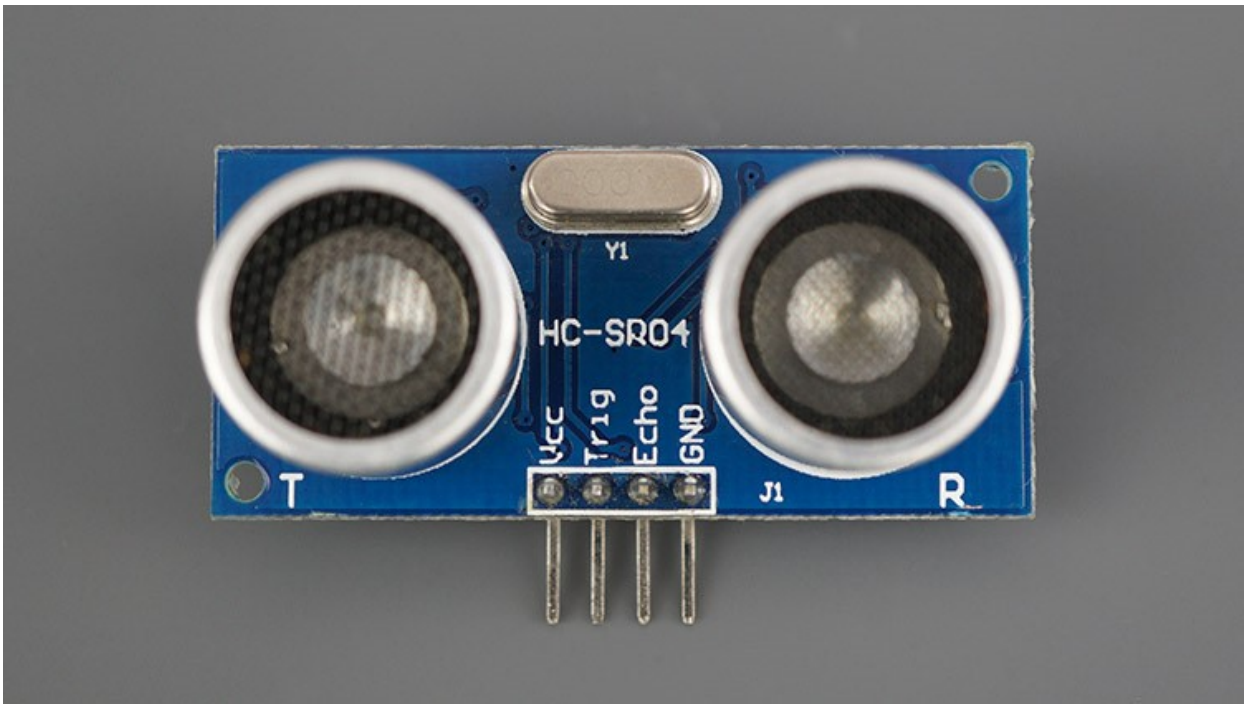
- ESP8266
- Ultra sonic sensor
- LEDs (optional)
- Resistors (optional)
- Some jumper wires (Female to Female)
- A small breadboard will be extremely good. We can do the work without it as well.
- Good microUSB **Data** cable (not charging cable)
- Laptop



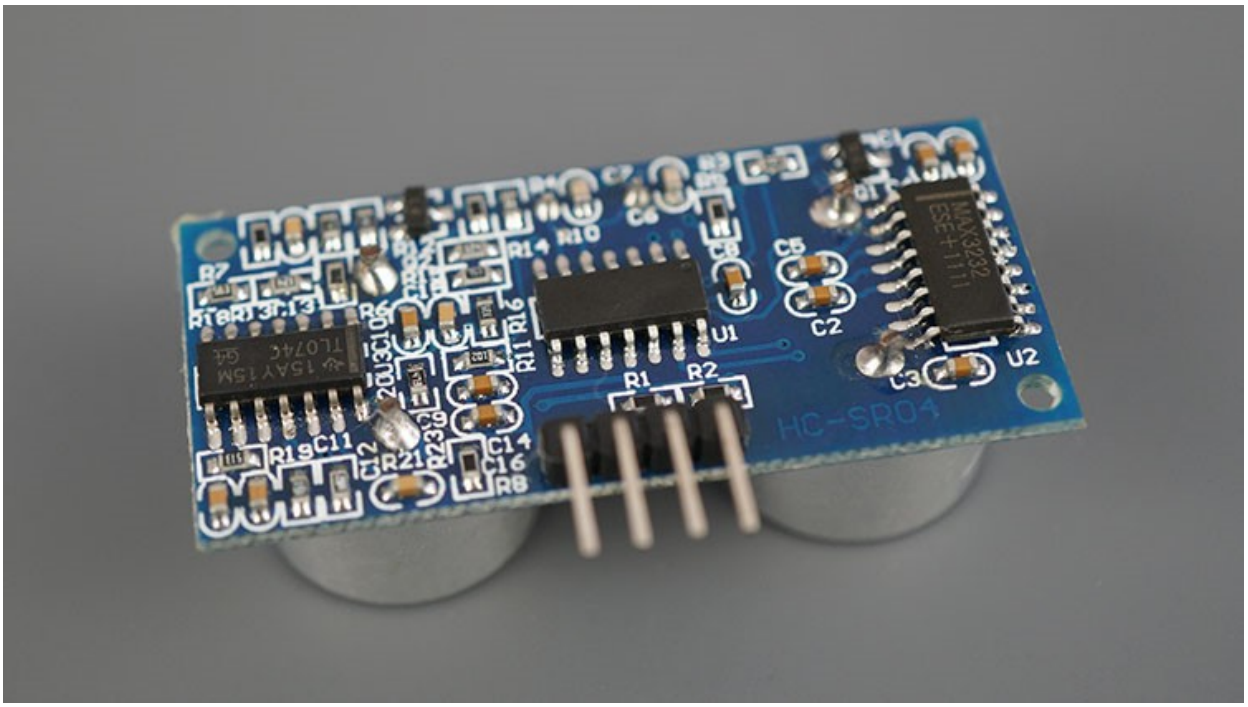
## Introducing the HC-SR04 Ultrasonic Sensor

The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an object. This sensor reads from 2cm to 400cm (0.8inch to 157inch) with an accuracy of 0.3cm (0.1inches), which is good for most hobbyist projects. In addition, this particular module comes with ultrasonic transmitter and receiver modules.

The following picture shows the HC-SR04 ultrasonic sensor.




The next picture shows the other side of the sensor.




## HC-SR04 Ultrasonic Sensor Technical Data

The following table shows the key features and specs of the HC-SR04 ultrasonic sensor. For more information, you should consult the sensor's datasheet.

| | |
|---|---|
| **Power Supply** | 5V DC |
| **Working Current** | 15 mA |
| **Working Frequency** | 40 kHz |
| **Maximum Range** | 4 meters |
| **Minimum Range** | 2 cm |
| **Measuring Angle** | 15º |
| **Resolution** | 0.3 cm |
| **Trigger Input Signal** | 10uS TTL pulse |
| **Echo Output Signal** | TTL pulse proportional to the distance range |
| **Dimensions** | 45mm x 20mm x 15mm |

## HC-SR04 Ultrasonic Sensor Pinout

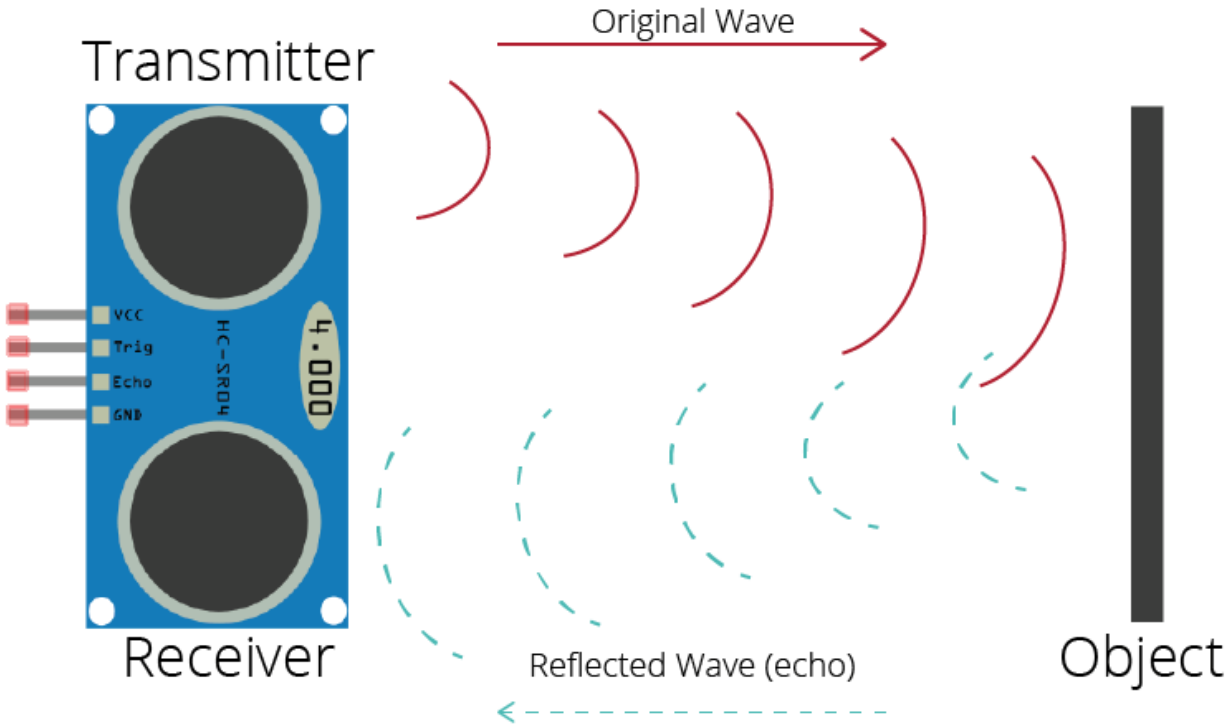
Here's the pinout of the HC-SR04 Ultrasonic Sensor.

| | |
|---|---|
| VCC | Powers the sensor (5V) |
| Trig | Trigger Input Pin |
| Echo | Echo Output Pin |
| GND | Common GND |

## How Does the HC-SR04 Ultrasonic Sensor Work?

The ultrasonic sensor uses sonar to determine the distance to an object. Here's how it works:

1. The ultrasound transmitter (trig pin) emits a high-frequency sound (40 kHz).
2. The sound travels through the air. If it finds an object, it bounces back to the module.
3. The ultrasound receiver (echo pin) receives the reflected sound (echo).

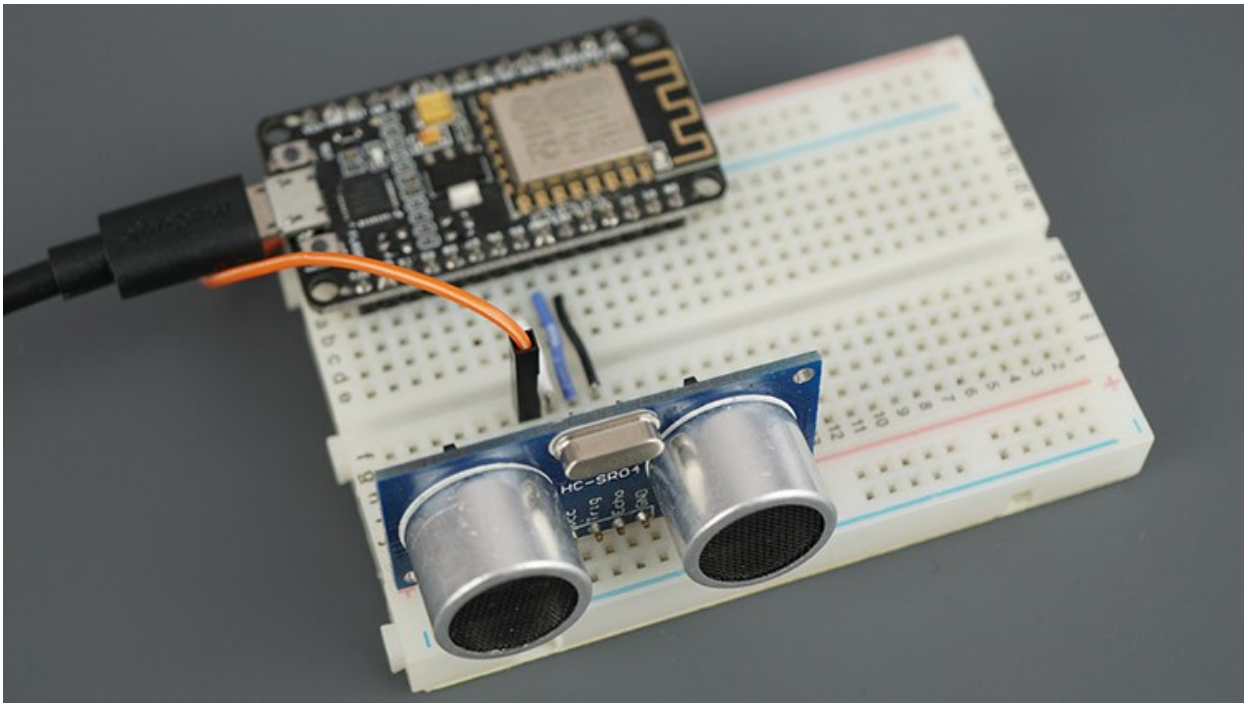



Taking into account the sound's velocity in the air and the travel time (time passed since the transmission and reception of the signal) we can calculate the distance to an object. Here's the formula:

distance to an object = (speed of sound in the air * (time/2))

speed of sound in the air at 20ºC (68ºF) = **343m/s**

## Parts Required

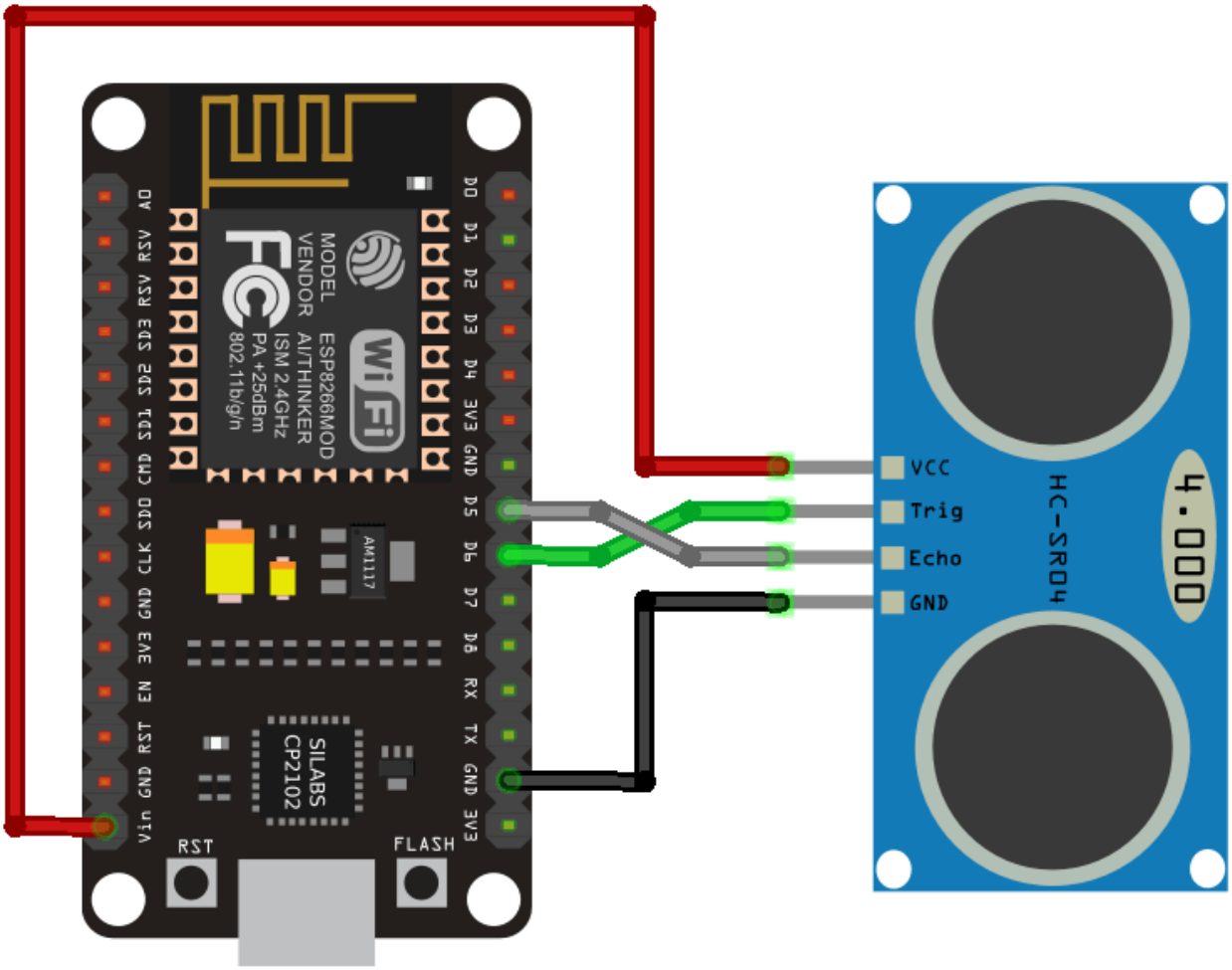



To complete this tutorial you need the following parts:

- HC-SR04 Ultrasonic Sensor
- ESP8266 (read Best ESP8266 development boards)
- Breadboard
- Jumper wires

## Schematic – ESP8266 NodeMCU with HC-SR04 Ultrasonic Sensor

Wire the HC-SR04 ultrasonic sensor to the ESP8266 as shown in the following schematic diagram. We're connecting the Trig pin to GPIO 5 and the Echo pin to GPIO 18, but you can use any other suitable pins.




| Ultrasonic Sensor | ESP8266 |
|---|---|
| VCC | VIN |
| Trig | GPIO 12 (D6) |
| Echo | GPIO 14 (D5) |
| GND | GND |

## How the Code Works

First, define the trigger and the echo pins.

```
const int trigPin = 12;
const int echoPin = 14;
```

In this example, we're using GPIO 12 and GPIO 14. But you can use any other suitable GPIOs—read ESP8266 Pinout Reference: Which GPIO pins should you use?

The `SOUND_SPEED` variable saves the velocity of sound in the air at 20ºC. We're using the value in cm/uS.

```
#define SOUND_SPEED 0.034
```

The `CM_TO_INCH` variable allows us to convert distance in centimeters to inches.

```
#define CM_TO_INCH 0.393701
```

Then, initialize the following variables.

```
long duration;
float distanceCm;
float distanceInch;
```

The `duration` variable saves the travel time of the ultrasonic waves (time elapsed since transmission and reception of the pulse wave).

The `distanceCm` and `distanceInch`, as the names suggest, save the distance to an object in centimeters and inches.

## setup()

In the `setup()`, initialize a serial communication at a baud rate of 115200 so that we can print the measurements on the Serial Monitor.

```
Serial.begin(115200); // Starts the serial communication
```

Define the trigger pin as an `OUTPUT`—the trigger pin emits the ultrasound. And define the echo pin as an `INPUT`—the echo pin receives the reflected wave and sends a signal to the ESP8266 that is proportional to the travel time.

```
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
```

## loop()

In the `loop()`, the following lines produce a 10uS `HIGH` pulse on the trigger pin—this means the pin will emit an ultrasound. Note that before sending the pulse, we give a short `LOW` pulse to ensure you'll get a clean `HIGH` pulse.

```
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

We use the `pulseIn()` function to get the sound wave travel time:

```
duration = pulseIn(echoPin, HIGH);
```

The pulseIn() function reads a HIGH or a LOW pulse on a pin. It accepts as arguments the pin and the state of the pulse (either HIGH or LOW). It returns the length of the pulse in microseconds. The pulse length corresponds to the time it took to travel to the object plus the time traveled on the way back.

Then, we simply calculate the distance to an object taking into account the sound speed.

```
distanceCm = duration * SOUND_SPEED/2;
```

Convert the distance to inches:

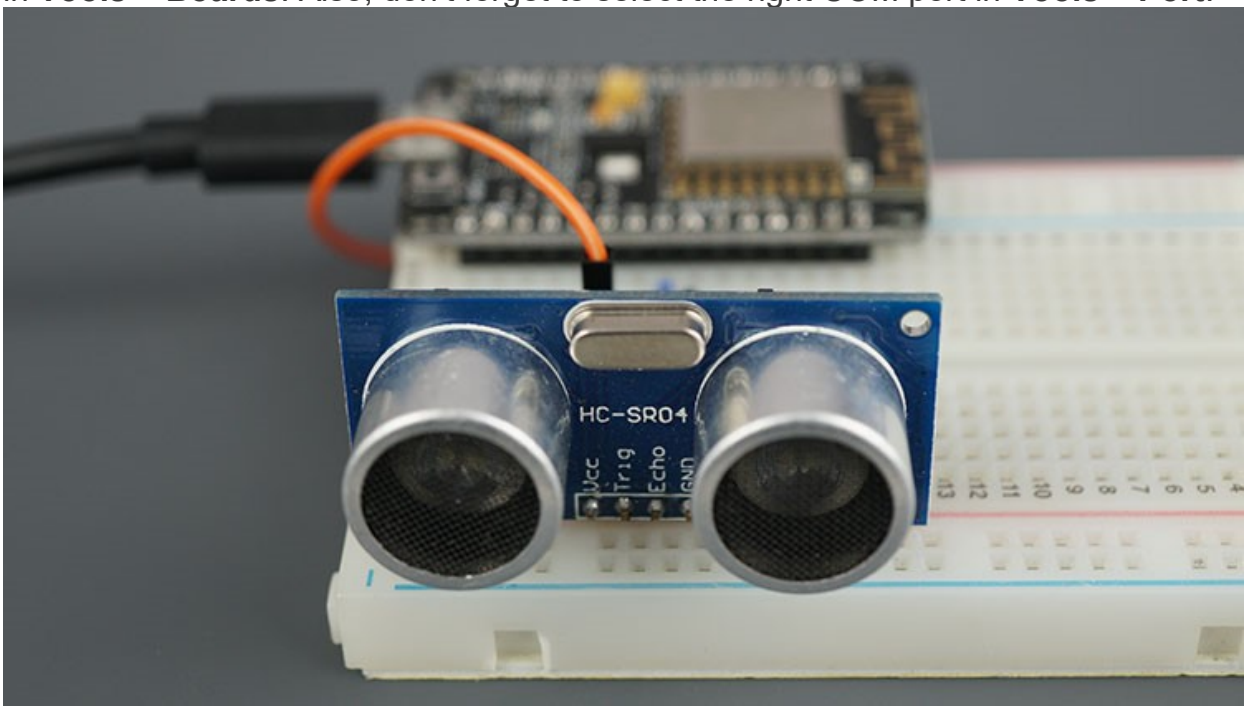
```
distanceInch = distanceCm * CM_TO_INCH;
```

And finally, print the results on the Serial Monitor.

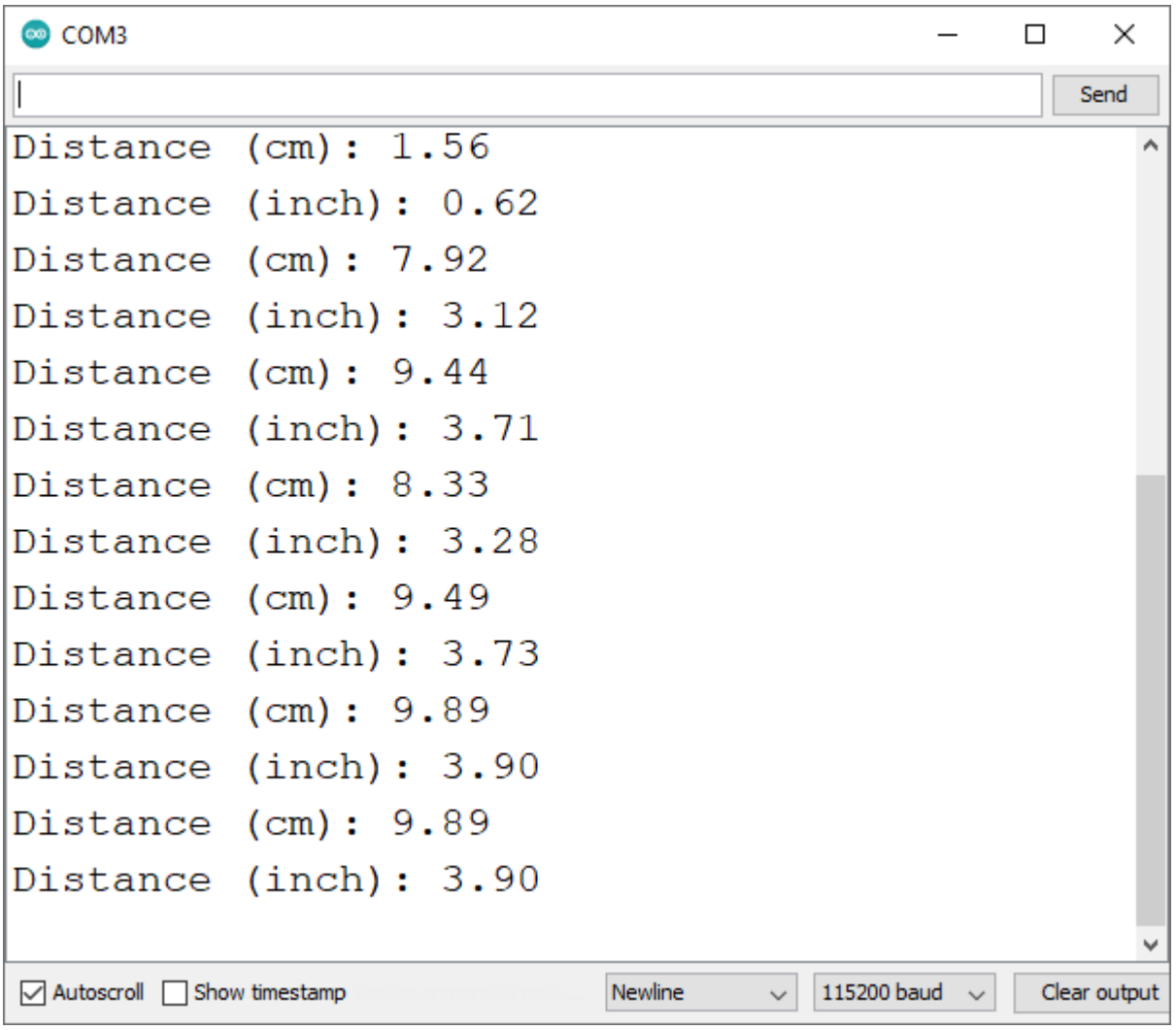
```
Serial.print("Distance (cm): ");
Serial.println(distanceCm);
Serial.print("Distance (inch): ");
Serial.println(distanceInch);
```

## Demonstration

Upload the code to your board. Don't forget to select the board you're using in **Tools** > **Boards**. Also, don't forget to select the right COM port in **Tools** > **Port**.




After uploading, open the Serial Monitor at a baud rate of 115200. Press the on-board RST button to restart the board and it will start printing the distance to the closest object on the Serial Monitor. Something as shown in the picture below.


COM3
Send
Distance (cm): 1.56
Distance (inch): 0.62
Distance (cm): 7.92
Distance (inch): 3.12
Distance (cm): 9.44
Distance (inch): 3.71
Distance (cm): 8.33
Distance (inch): 3.28
Distance (cm): 9.49
Distance (inch): 3.73
Distance (cm): 9.89
Distance (inch): 3.90
Distance (cm): 9.89
Distance (inch): 3.90
Autoscroll
Show timestamp
Newline
115200 baud
Clear output