

# ULTRASONIC SENSOR INTERFACED WITH NODEMCU

HC-SR04 ultrasonic sensor is composed of ultrasonic transmitter, ultrasonic receiver and a control circuit. HC-SR04 ultrasonic transmitter transmits ultrasound waves at 40,000 Hz. Transmitted waves bounce back if they hit any flat surface/object in their path. Bounced back waves reaches the ultrasonic receiver. Ultrasonic receiver receives the bounced back waves and notifies the control circuit about it. Control circuit then calculates the time taken by waves to reach back after transmission. Time is then manipulated to approximate the distance travelled by waves or what is the distance between the sensor and the object? from which ultrasound waves bounced back. HC-SR04 can measure distance between an active range of 2 cm to 4 meters. HC-SR04 requires 5 volts and 15 mA of power for operation. HC-SR04 has four pins. Two are power pins. Vcc is +ve pin apply 5v to this pin and Gnd is ground pin connect -ve of 5v power source with it. The other two pins are **Trigger** and **Echo**.

- **Trigger pin** is triggered by external controller to outburst an ultrasound wave.
- **Echo pin** notifies external controller when receiver receives back the bounced back wave.

Time is converted in to distance using the speed of sound in air formula. According to universal speed of sound in air formula

$$\text{Time} = \text{Distance} / \text{Speed}$$

Where **Speed = speed of sound in air. Which is 340 m/s**

Hence reorganizing the formula **Distance= Time \* 340(speed of sound in air).**

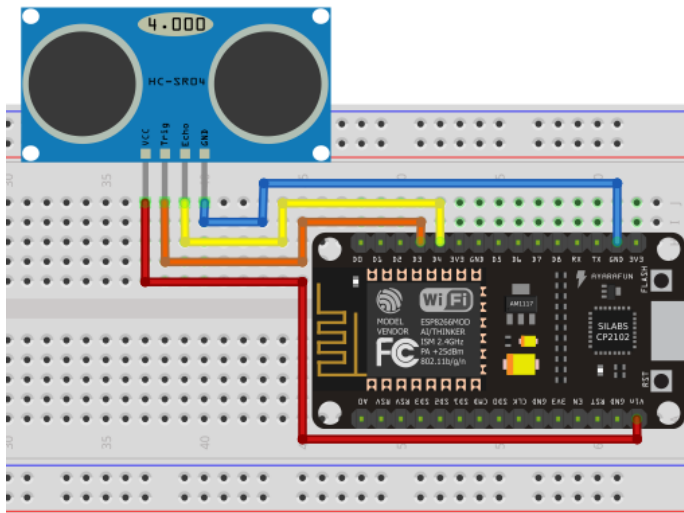
## Hardware Requirements:

- NodeMCU
- HC-SR04 (Ultra-sonic Sensor)
- Bread Board
- Jumper Wires
- Micro USB Cable

## Software Requirements:

➤ Arduino IDE

## Circuit Diagram:



NodeMCU	HC-SR04
Vin	Vcc
D3	Trig
D4	Echo
GND	GND

## Code:

```
#include <ESP8266WiFi.h>

const int trigP = D3;    //D4
const int echoP = D4;    //D3

long duration;
int distance;

void setup() {
  pinMode(trigP, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoP, INPUT);  // Sets the echoPin as an Input
  Serial.begin(9600);     // Open serial channel at 9600 baud
  rate
}

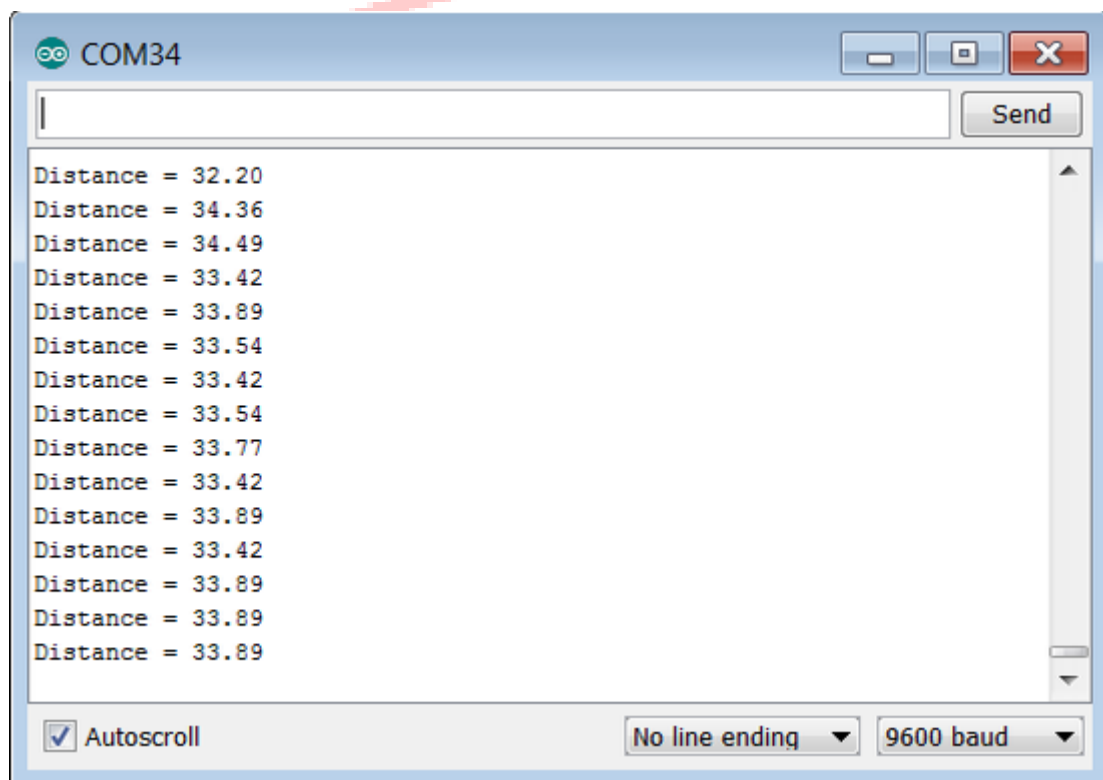
void loop() {
  digitalWrite(trigP, LOW); // Makes trigPin low
  delayMicroseconds(2);     // 2 micro second delay

  digitalWrite(trigP, HIGH); // trigPin high
  delayMicroseconds(10);    // trigPin high for 10 micro
  seconds
  digitalWrite(trigP, LOW); // trigPin low
```

```
duration = pulseIn(echoP, HIGH);    //Read echo pin, time in
microseconds
distance= duration*0.034/2;         //Calculating actual/real
distance

Serial.print("Distance = ");        //Output distance on
arduino serial monitor
Serial.println(distance);
delay(3000);                        //Pause for 3 seconds and
start measuring distance again
}
```

### Output:



iSMRITI