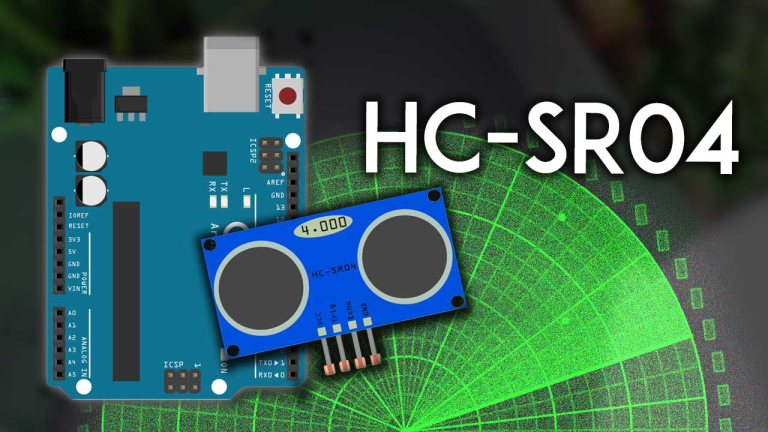
**Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino**

[**https://rand** **omnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/**](https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/)

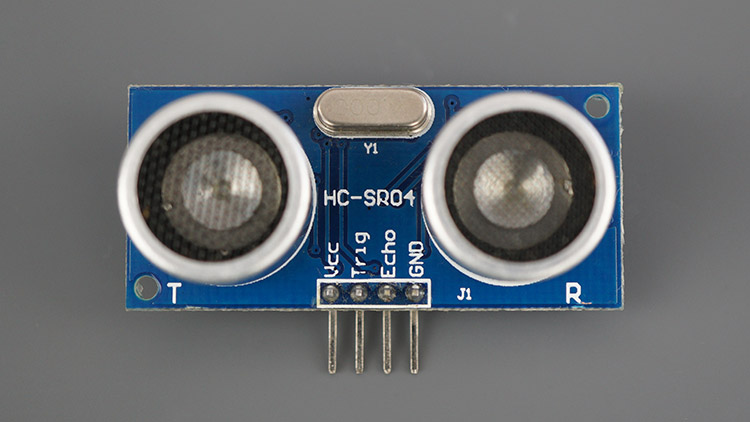
This article is a guide about the Ultrasonic Sensor HC-SR04. We’ll explain how it works, show you some of its features and share an Arduino project example you can follow to integrate into your projects. We provide a schematic diagram on how to wire the ultrasonic sensor and an example sketch with the Arduino.



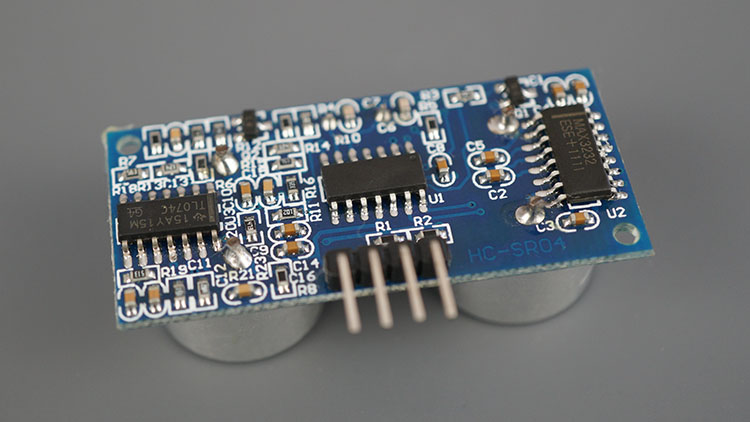
### **Description**

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.



### **Features**

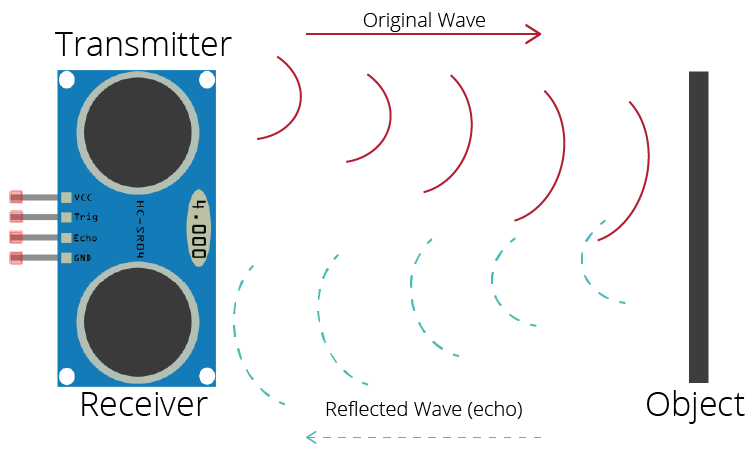
Here’s a list of some of the HC-SR04 ultrasonic sensor features and specs—for more information, you should consult the sensor’s datasheet:

* Power Supply :+5V DC
* Quiescent Current : <2mA
* Working Current: 15mA
* Effectual Angle: <15°
* Ranging Distance : 2cm – 400 cm/1″ – 13ft
* Resolution : 0.3 cm
* Measuring Angle: 30 degree
* Trigger Input Pulse width: 10uS TTL pulse
* Echo Output Signal: TTL pulse proportional to the distance range
* Dimension: 45mm x 20mm x 15mm

### **How Does it Work?**

The ultrasonic sensor uses sonar to determine the distance to an object. Here’s what happens:

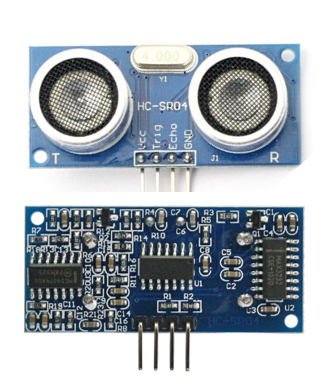
1. The ultrasound transmitter (trig pin) emits a high-frequency sound (40 kHz).
2. The sound travels through the air. If it finds an object, it bounces back to the module.
3. The ultrasound receiver (echo pin) receives the reflected sound (echo).



he time between the transmission and reception of the signal allows us to calculate the distance to an object. This is possible because we know the sound’s velocity in the air. Here’s the formula:

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

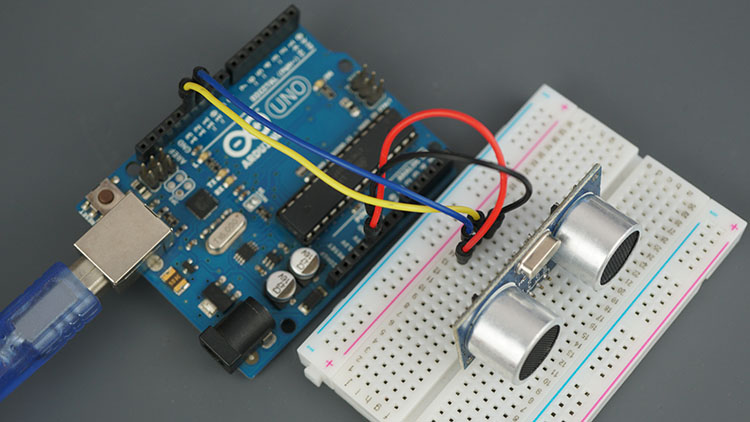
### **HC-SR04 Ultrasonic Sensor Pinout**



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 |

## **Arduino with HC-SR04 Sensor**



This sensor is very popular among Arduino tinkerers. So, here we provide an example of how to use the HC-SR04 ultrasonic sensor with the Arduino. In this project, the ultrasonic sensor reads and writes the distance to an object in the serial monitor.

The goal of this project is to help you understand how this sensor works. Then, you should be able to use this example in your own projects.

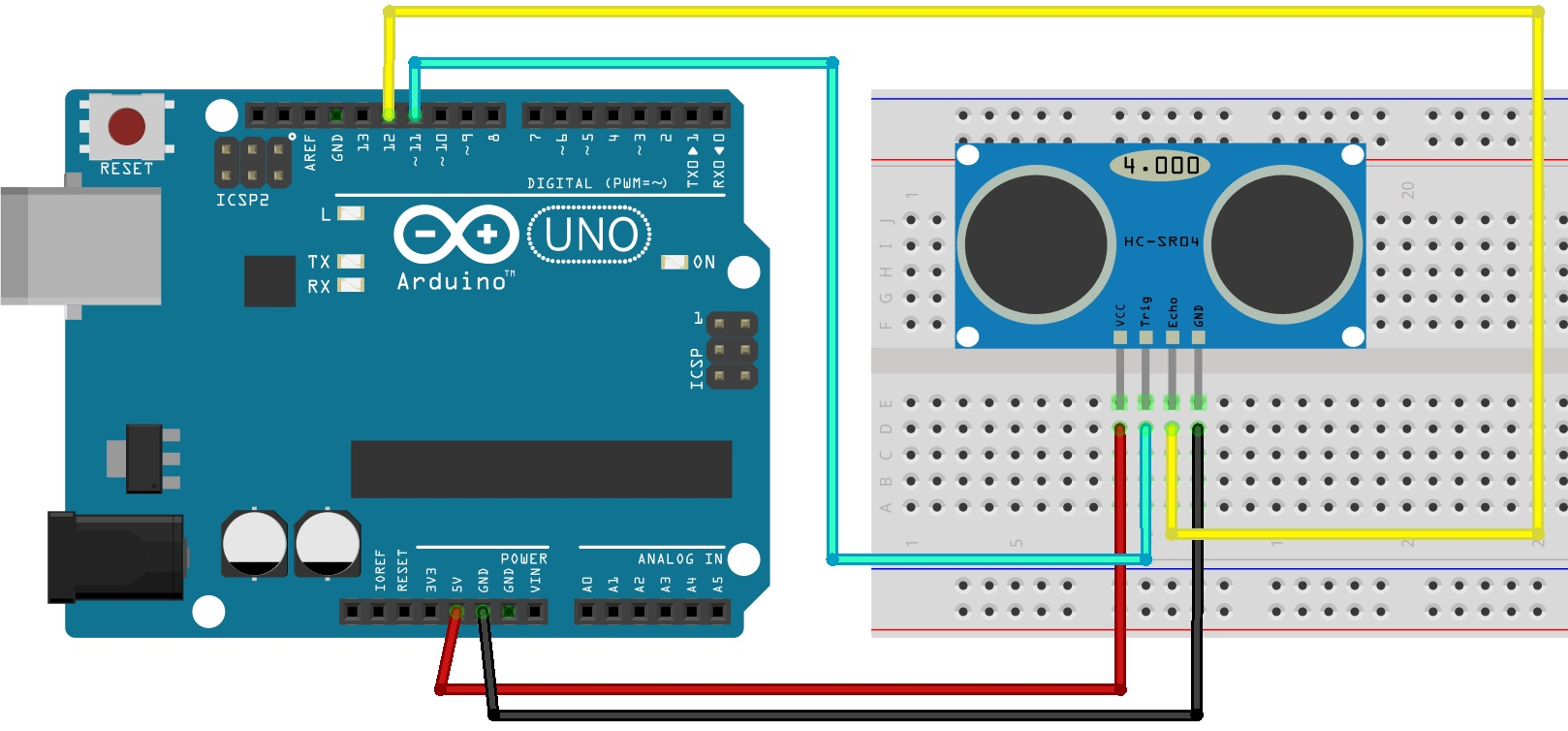
**Parts Required**

Here’s a list of the parts required to follow the next tutorial:

* [Arduino UNO](https://makeradvisor.com/tools/compatible-arduino-uno-r3-board/)
* [Ultrasonic Sensor (HC-SR04)](https://makeradvisor.com/tools/ultrasonic-sensor-hc-sr04/)
* [Breadboard](https://makeradvisor.com/tools/mb-102-solderless-breadboard-830-points/)
* [Jumper wires](https://makeradvisor.com/tools/jumper-wires-kit-120-pieces/)

## **Arduino with HC-SR04 Sensor Wiring**

Follow the next schematic diagram to wire the HC-SR04 ultrasonic sensor to the Arduino.



**Code**

(raw code - <https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/Ultrasonic_Sensor_HC-SR04.c>)

/\*

\* created by Rui Santos, https://randomnerdtutorials.com

\*

\* Complete Guide for Ultrasonic Sensor HC-SR04

\*

Ultrasonic sensor Pins:

VCC: +5VDC

Trig : Trigger (INPUT) - Pin11

Echo: Echo (OUTPUT) - Pin 12

GND: GND

\*/

int trigPin = 11; // Trigger

int echoPin = 12; // Echo

long duration, cm, inches;

void setup() {

//Serial Port begin

Serial.begin (9600);

//Define inputs and outputs

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

}

void loop() {

// The sensor is triggered by a HIGH pulse of 10 or more microseconds.

// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:

digitalWrite(trigPin, LOW);

delayMicroseconds(5);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Read the signal from the sensor: a HIGH pulse whose

// duration is the time (in microseconds) from the sending

// of the ping to the reception of its echo off of an object.

pinMode(echoPin, INPUT);

duration = pulseIn(echoPin, HIGH);

// Convert the time into a distance

cm = (duration/2) / 29.1; // Divide by 29.1 or multiply by 0.0343

inches = (duration/2) / 74; // Divide by 74 or multiply by 0.0135

Serial.print(inches);

Serial.print("in, ");

Serial.print(cm);

Serial.print("cm");

Serial.println();

delay(250);

}