Introduction:

The main aim of the system is to detect object by the help of Ultrasonic wave. Ultrasonic wave is a sound wave transmitted at a frequency greater than 20,000 Hz, or beyond the normal hearing range of humans.

Components used:

1. Arduino Uno
2. HC-SR04 Ultrasonic sensor
3. A box for detection testing
4. Jumper wires
5. Breadboard

Arduino Uno:

**Arduino Uno** is a microcontroller board based on the Microchip ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

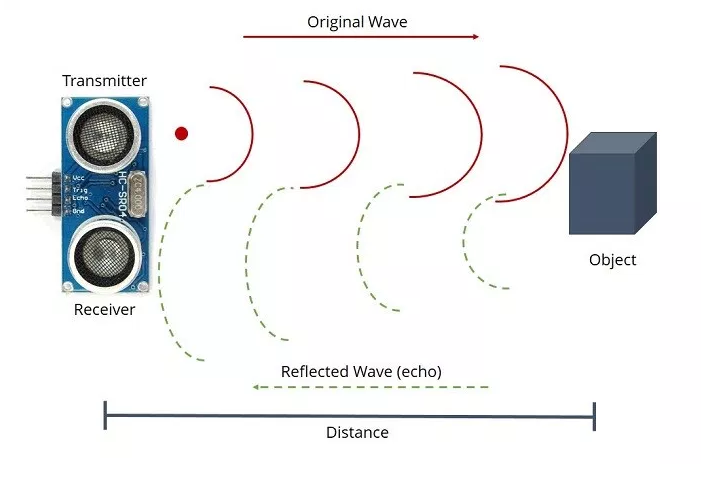
HC-SR04 Ultrasonic sensor:

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm or 1” to 13 feet. It comes complete with ultrasonic transmitter and receiver module.

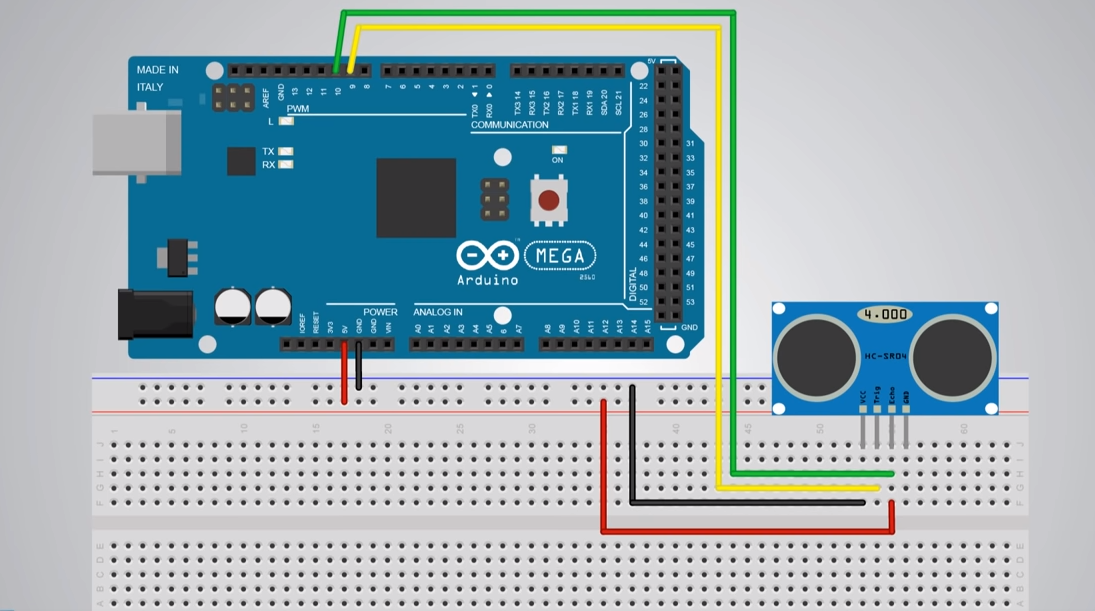
How Does it Work?

The ultrasonic sensor uses ultrasonic wave to determine the distance to an object. Here’s what happens:  
 1) The transmitter sends a signal: a high-frequency sound  
 2) When the signal finds an object, it is reflected  
 3) Then echo pin receives it.

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



Circuit Diagram:



Code:

const int trigPin=9;

const int echoPin=10;

long duration;

int distance;

void setup() {

// put your setup code here, to run once:

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(12,OUTPUT);

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance= duration\*0.034/2;

Serial.print("Distance: ");

Serial.println(distance);

if(distance<=5){

digitalWrite(12, HIGH);

}

else{

digitalWrite(12,LOW);

}

}

How Does the Code Work:

At first, the two pins 9 and 10 are set to variables trigPin and echoPin respectively.

**const int trigPin=9;**

**const int echoPin=10;**

Then two variables are initialized.

**long duration;**

**int distance*;***

In the void setup() function, three pins are setup either INPUT or OUTPUT according to their use.

**void setup() {**

**pinMode(trigPin, OUTPUT);**

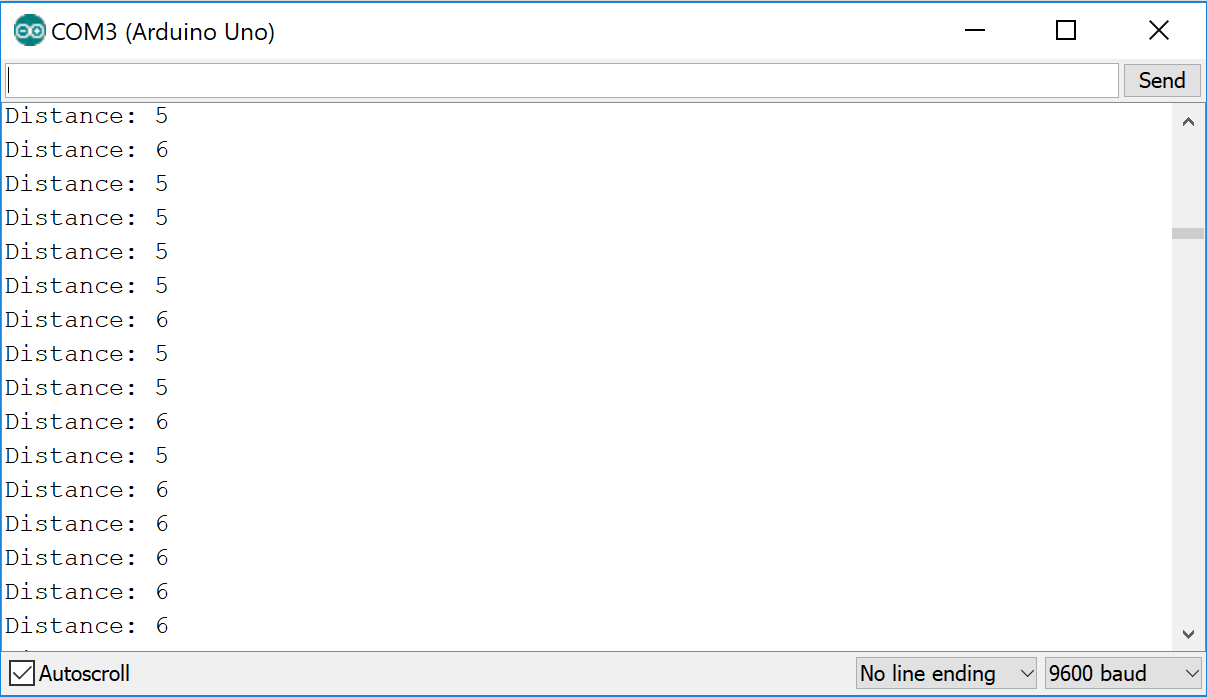
**pinMode(echoPin, INPUT);**

**pinMode(12,OUTPUT);**

**Serial.begin(9600);**

**}**

Serial.begin(9600) is used for showing the results on the serial monitor like the picture.

**

**void loop() {**

**// put your main code here, to run repeatedly:**

**digitalWrite(trigPin, LOW;**

**delayMicroseconds(2);**

**digitalWrite(trigPin, HIGH);**

**delayMicroseconds(10);**

**digitalWrite(trigPin, LOW);**

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

**distance= duration\*0.034/2;**

**Serial.print("Distance: ");**

**Serial.println(distance);**

**if(distance<=5){**

**digitalWrite(12, HIGH);**

**}**

**else{**

**digitalWrite(12,LOW);**

**}**

In the void loop() function, the trigPin is set low for 2microseconds to make the trigPin clear. Afterwards, for generating the ultrasound wave we set the trigPin to a HIGH state. Then the trigPin is set LOW.

Then we read the time travelled using the pulseIn function and save it in the duration variable. Distance= velocity x time. So we can calculate distance by using the speed of light here and then dividing the distance by 2 as wave travels the distance twice. So, distance= duration \* 0.034 / 2. Then Serial communication is used to see the distance of the object. At last, the if condition is written to set pin 12 HIGH, which is connected to a LED, when the distance is less than or equal to 5cm.

