

HC-SR04 ULTRASONIC DISTANCE SENSOR INTERFACING WITH ARDUINO

Ultrasonic Sensor

Ultrasonic sensor is used in various applications which needs the information regarding the distance of the nearby obstacle from the sensor module.

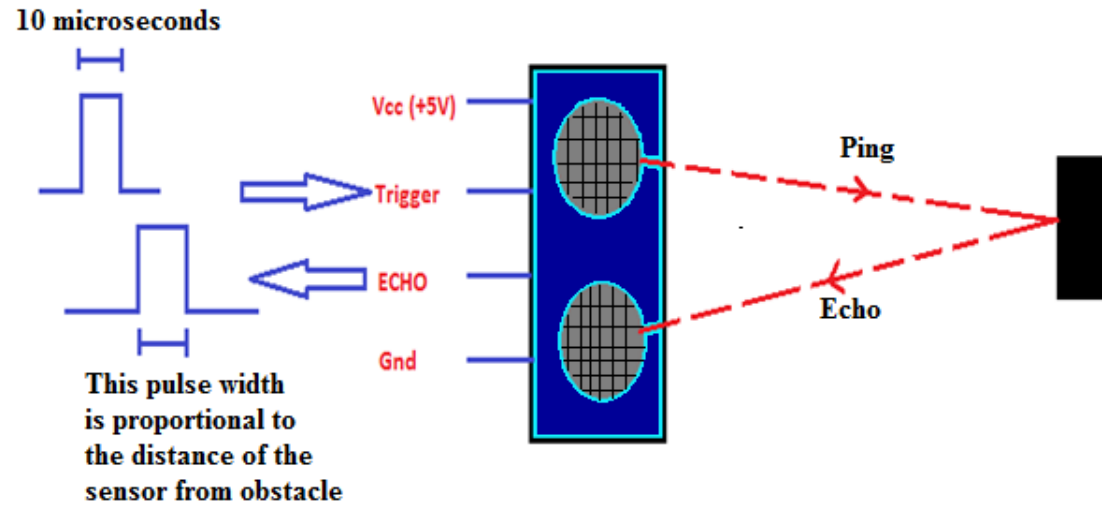
This is used in robotics applications.

The self-driving cars also make use of this technology and decision regarding car speed, applying brakes are made depending on the distance of the car from the nearby vehicle.

It is also used in cars as a parking sensor.

Ultrasonic sensor module, HC-SR04 that can be easily interfaced with Arduino Uno board for measuring the distance from the nearby obstacle.

Distance Calculation



Event Sequence

Trigger pin is given a pulse of minimum 10 microseconds HIGH time.

Module will send eight pulses of 40 KHz frequency in the nearby region and also make the state of ECHO pin logic HIGH.

The ECHO pin status goes to logic LOW level when the ultrasonic waves return back from the nearby obstacle.

The pulse width of ECHO signal is proportional to the distance of the obstacle from the sensor module.

The distance of the obstacle from the sensor is obtained from the time duration for which the ECHO pin status has remained at logic HIGH level.

Distance Formula

$$\text{Distance} = (\text{speed} * \text{Time}) / 2$$

Speed of ultrasonic waves is approximately 340 m/s = 340 x 100 cm/s

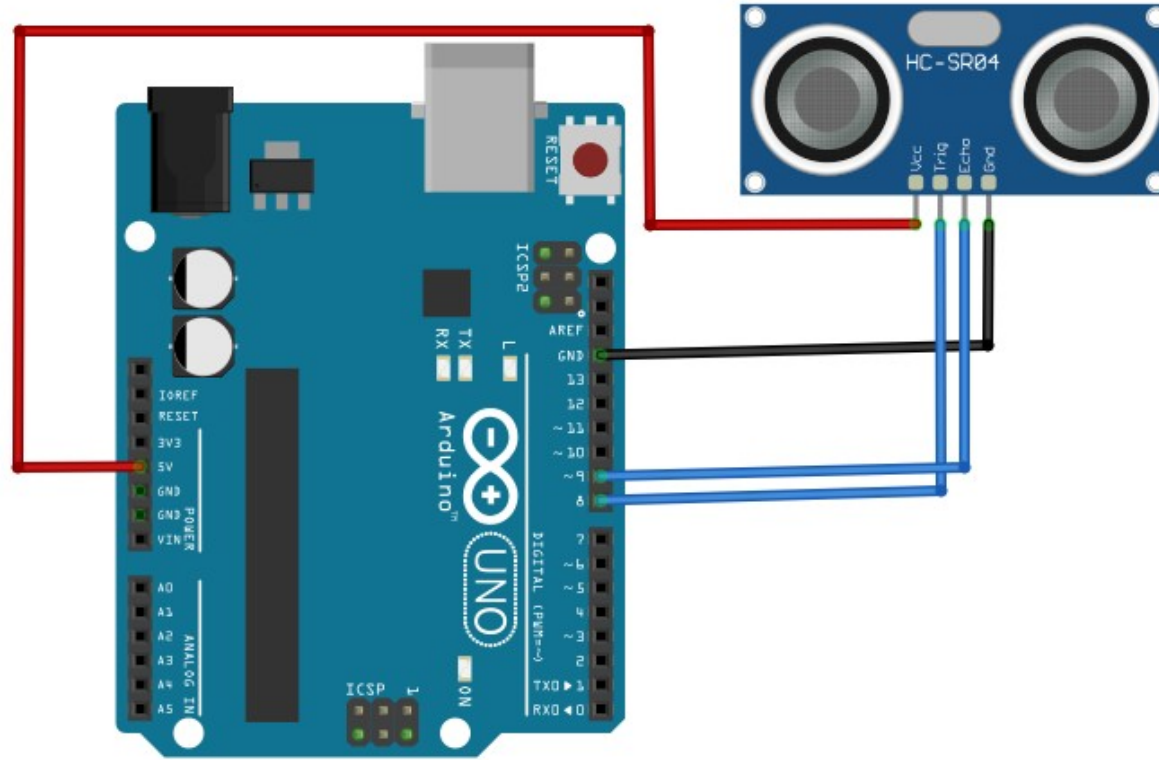
Time can be specified as: (duration * 10⁻⁶) sec

So, the Distance formula becomes:

$$\text{Distance} = (340 * 100 * \text{duration} * 10^{-6}) / 2$$

$$\text{Distance} = (\text{duration} / 58.6) \text{ cm}$$

Circuit Diagram



Code

```
float duration; // variable which will hold time duration
float distance; // variable which will hold the distance in cms
int trigger = 8; // trigger pin is connected on pin 8
int echo = 9; // echo pin is connected to pin 9
int led = 13; // led is connected to pin 13
void setup()
{
  pinMode(led, OUTPUT); // configure led as o/p pin
  pinMode(trigger, OUTPUT); // trigger pin configured as o/p
  pinMode(echo, INPUT); // echo pin configured as i/p
  Serial.begin(9600);
  // initialize serial communication at 9600 bps baud rate
}
```

```
void loop()
{
  // apply a pulse on trigger pin
  digitalWrite(trigger, LOW); // make trigger LOW
  delayMicroseconds(2); // delay of 2 microseconds
  digitalWrite(trigger, HIGH); // make trigger HIGH
  delayMicroseconds(10); // delay of 10 microseconds
  digitalWrite(trigger, LOW); // make trigger LOW
  duration = pulseIn(echo, HIGH);
  // calculate time for which echo pin remains HIGH
  distance = duration / 58.6;
  // calculate distance in cm using this formula
  Serial.println(distance);
  // print distance from obstacle on serial port
}
```

Code Contd.

```
if (distance < 30) // check if distance is less than 30cm
{
  digitalWrite(led, HIGH); // turn ON the led
}
else
{
  digitalWrite(led, LOW); // turn OFF the led
}
delay(100); // give delay of 100 msec for the next scan to begin
}
```


O/P on Serial Monitor

