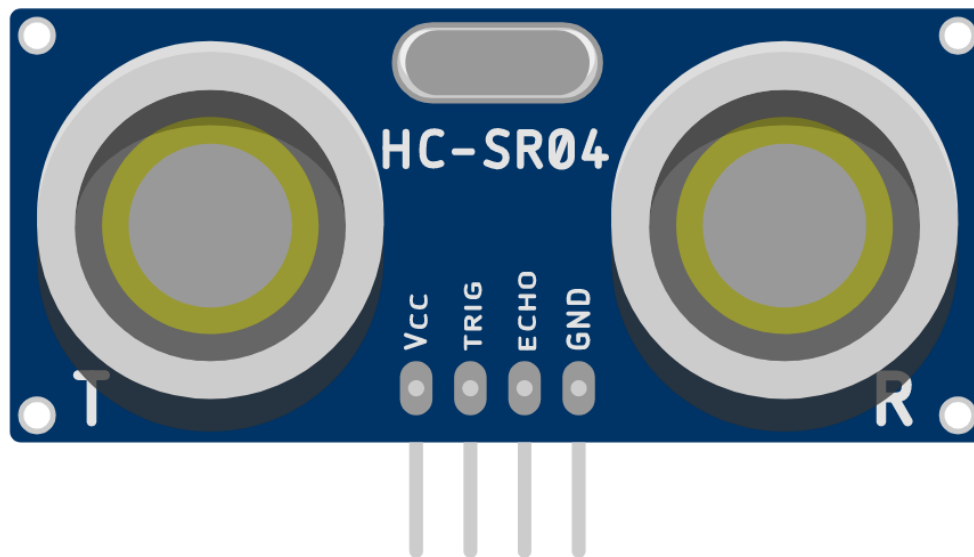


Ultrasonic Distance Sensor HC-SR04



What is ultrasonic distance sensor?

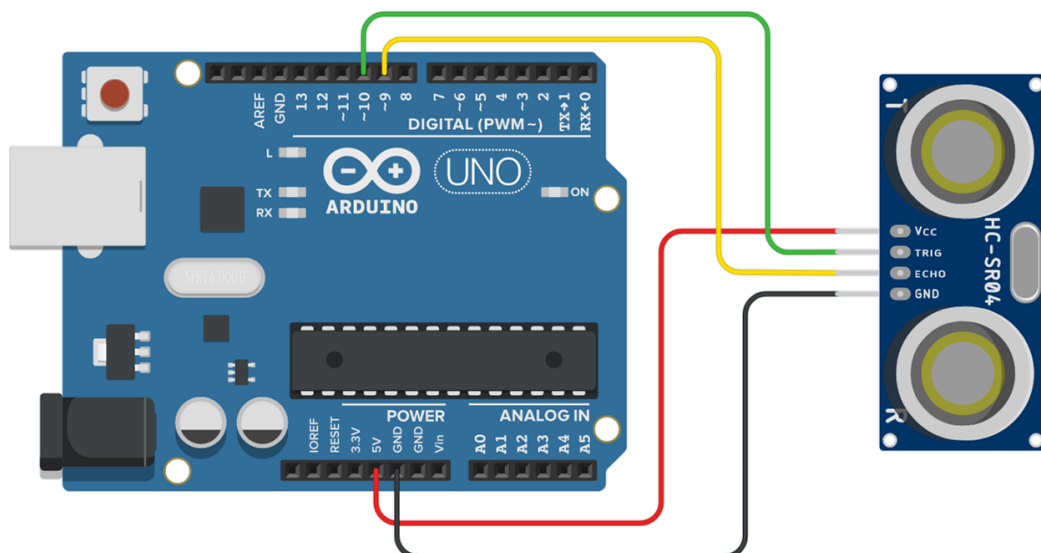
- It measures distance by sending ultrasound (40 kHz) and this ultrasound will bounce off the obstacle and return to the sensor.
- Distance is measured by time it takes the ultrasound to travel from the sensor and back to the sensor.
 - **distance = (travel time × 340 m/s*) / 2**
 - * *speed of sound*

Electric parameters

- Working voltage DC 5V
- Working current: 15 mA
- Working frequency: 40 kHz
- Max range: 4 m
- Min range: 2 cm
- Measuring Angle: 0-15°

Wire connections:

- 5V supply → VCC
- 0V GND → GND
- Trigger Pulse input: Dn → TRIG
- Echo Pulse output: Dm → ECHO



Code

This program will use the ultrasonic distance sensor to read the distance of the object. Under certain distance a LED will turn on.

```
/* define pins */
const int trigPin = 10;           // digital output for TRIGGER pin (start)
const int echoPin = 9;            // digital output for ECHO pin (return)
const int LED = 6;                // digital output for LED
long duration;                   // time it takes to travel from trigPin to echoPin
int distance;                     // distance of the object

void setup() {
    pinMode(trigPin, OUTPUT);      // trigger pin is always output
    pinMode(echoPin, INPUT);       // echo pin is always input
    pinMode(LED, OUTPUT);
}

// set data reading for communicating with Serial Monitor
Serial.begin(9600);
}

void loop() {
    /* first disable/reset the trigger pin to 0/LOW for two microsecond before
    beginning the measurement */
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    /* send an ultrasonic wave for 10 microseconds */
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

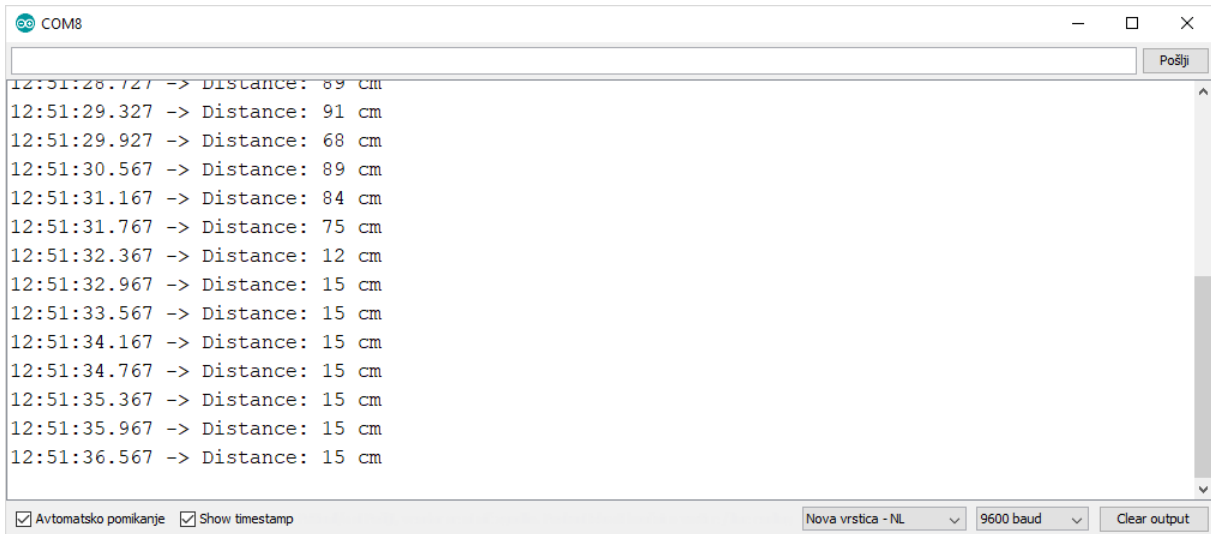
    /* with function pulseIn we will wait for the wave from trigPin to arrive
    and set it to HIGH */
    duration = pulseIn(echoPin, HIGH);

    /* calculation for distance (0.034 is the speed of sound in
    cm/microsecond) */
    distance = duration*0.034/2;

    /* Displaying distance with Serial Monitor */
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");
    delay(500);

    /* Do something useful with our measured distance.
    For example: turn on LED when object is too close
    */
    if (distance < 10) {
        digitalWrite(LED, HIGH);
    }
    else {
        digitalWrite(LED, LOW);
        delay(100);
    }
}
```

Serial monitor (CTRL+SHIFT+M)



The screenshot shows the Serial Monitor window for COM8. The output displays a series of distance measurements in centimeters, each preceded by a timestamp. The measurements start at 89 cm and then drop to 15 cm. The window includes a 'Pošlji' button and checkboxes for 'Avtomatsko pomikanje' and 'Show timestamp'. The bottom status bar shows 'Nova vrstica - NL', '9600 baud', and a 'Clear output' button.

```
12:51:28.727 -> Distance: 89 cm
12:51:29.327 -> Distance: 91 cm
12:51:29.927 -> Distance: 68 cm
12:51:30.567 -> Distance: 89 cm
12:51:31.167 -> Distance: 84 cm
12:51:31.767 -> Distance: 75 cm
12:51:32.367 -> Distance: 12 cm
12:51:32.967 -> Distance: 15 cm
12:51:33.567 -> Distance: 15 cm
12:51:34.167 -> Distance: 15 cm
12:51:34.767 -> Distance: 15 cm
12:51:35.367 -> Distance: 15 cm
12:51:35.967 -> Distance: 15 cm
12:51:36.567 -> Distance: 15 cm
```

This is the output of distance that is calculated based on the measured time.