

# ULTRASONIČNI SENZOR



HC-SR04

# HC-SR04 ULTRASONIC SENSOR

- jeftin sensor rastojanja
- jednostavan za upotrebu,
- domet mu se kreće od 2 do 400cm,
- u robotici se koristi za izbjegavanje prepreka,
- koristi se i u raznim projektima automatizacije,
- automobili koriste sličan sensor u sistemima za pomoć pri parkiranju.
- u nekim projektima potrebno je koristiti vodootporni sensor kao JSN-SR04T ili IR senzor koji nije osjetljiv na promjene temperature.



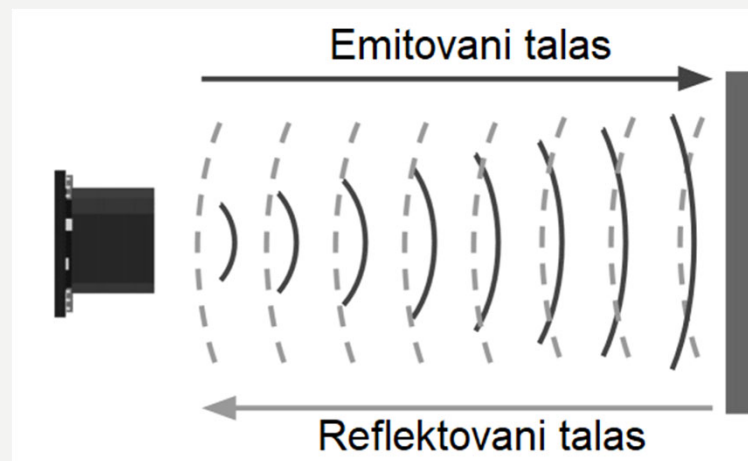
# KAKO SENZOR RADI?

Ultrasonični senzor emituje zvučni talas previsoke frekvencije da bi ga ljudsko uho čulo.

Talas putuje kroz vazduh, brzinom, približno 343 m/s.

Ako postoji objekat ispred senzora, zvučni talas će se reflektovati ka prijemu koji će ga detektovati.

Mjerenjem proteklog vremena od emitovanja do prijema talasa, može se izračunati rastojanje između senzora i objekta.



## ODREĐIVANJE RASTOJANJA?

- Na 20°C brzina zvuka je 343 m/s ili 0.034 cm/μs. Neka je proteklo vrijeme 2000μs. Ako se pomnoži brzina prostiranja i vrijeme putovanja talasa, dobija se rastojanje koje je talas prešao.

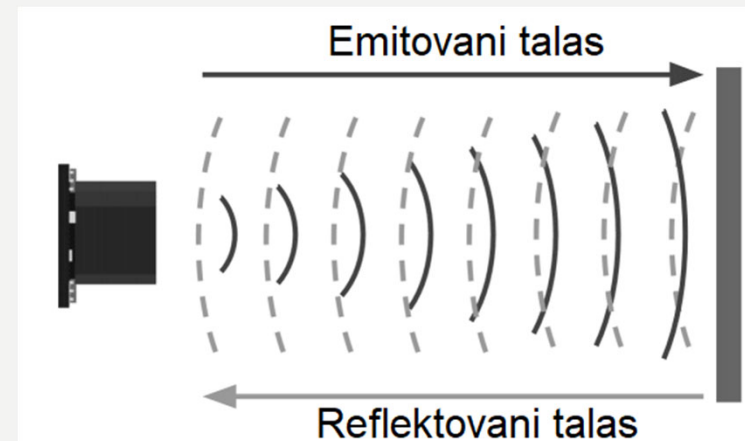
**Rastojanje = Brzina x Vrijeme**

- Međutim to nije rastojanje senzora i objekta. Ono je samo polovina toga.

**Rastojanje (cm) = Brzina zvuka (cm/μs) × Vrijeme (μs) / 2**

- Na kraju se ima:

**Rastojanje (cm) = 0.0343 (cm/μs) × 2000 (mple this μs) / 2 = 34.3 cm**



# ZAVISNOST BRZINE PROSTIRANJA ZVUKA OD TEMPERATURE VAZDUHA

- Brzina zvuka je veoma zavisna od temperature i u manjoj mjeri od vlažnosti vazduha. Zavisnost od temperature je približno 0.6 m/s po stepeni Celsius-a. Na 20°C može se uzeti da je brzina 343 m/s, ali ako se želi tačniji rezultat, potrebno je izračunati brzinu zvuka po formuli:

$$V \text{ (m/s)} = 331.3 + (0.606 \times T)$$

**V = Brzina zvuka (m/s)**

**T = Temperatura vazduha (°C)**

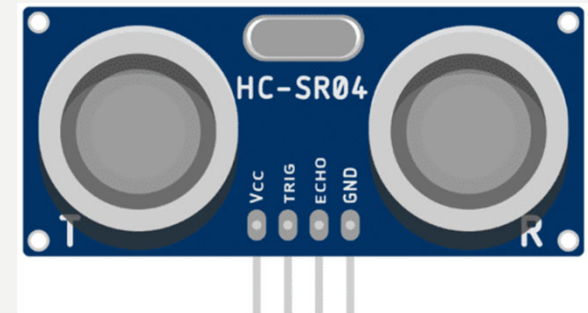
- Nije uzeta u obzir vlažnost, jer je njen uticaj relativno mali.

# KAKO SE UPRAVLJA HC-SR04 SENZOROM?

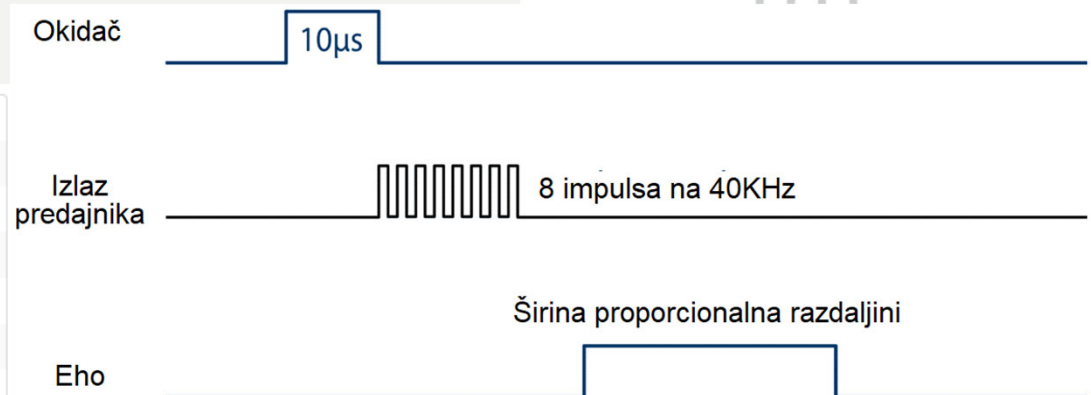
- Dva srebrna cilindra (ultrasonični primopredajnik)
- Jedan emituje zvučni talas a drugi ga prima.
- Za start emitovanja: na Trig pin impuls trajanja najmanje 10  $\mu$ s.
- Senzor zatim kreira 8 perioda zvučnog talasa na 40 kHz.
- Talas putuje, odbija se od prepreke i vraća nazad ka prijemniku.
- Na Eho pin-u se zatim pojavljuje impuls čije je trajanje proporcionalno rastojanju senzora od objekta.

Može se koristiti `pulseIn()` funkcija u Arduino kodu za očitavanje dužine impulse.

Nakon toga već datom formulom se izračunava rastojanje.

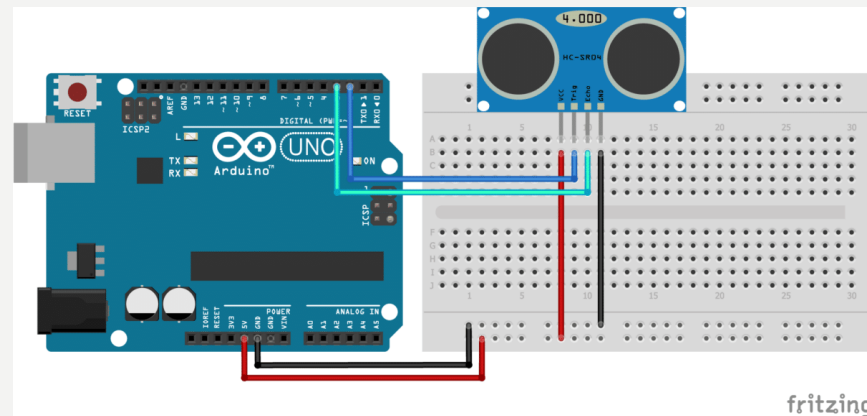


Operating voltage	5 V
Operating current	15 mA
Frequency	40 kHz
Measuring range	2 – 400 cm
Resolution	3 mm
Measuring angle	15 degrees
Trigger input signal	10 $\mu$ s high pulse



# POVEZIVANJE SENZORA SA ARDUINO UNO

Povezivanje senzora sa Arduino UNO razvojnom pločom je dato na slici:



Na slici su upotrijebljeni pinovi 2 and 3 za Trig i Echo pin, ali se može koristiti i bilo koji drugi digitalin pin.

HC-SR04	Arduino
VCC	5 V
Trig	Pin 2
Echo	Pin 3
GND	GND

# PRIMJER – ODREĐIVANJE RASTOJANJA OBJEKTA

```
// Define Trig and Echo pin:  
#define trigPin 2  
#define echoPin 3
```

```
// Define variables:  
long duration;  
int distance;
```

```
void setup() {  
  // Define inputs and outputs:  
  pinMode(trigPin, OUTPUT);  
  pinMode(echoPin, INPUT);
```

```
  //Begin Serial communication at a baudrate of 9600:  
  Serial.begin(9600);  
}
```

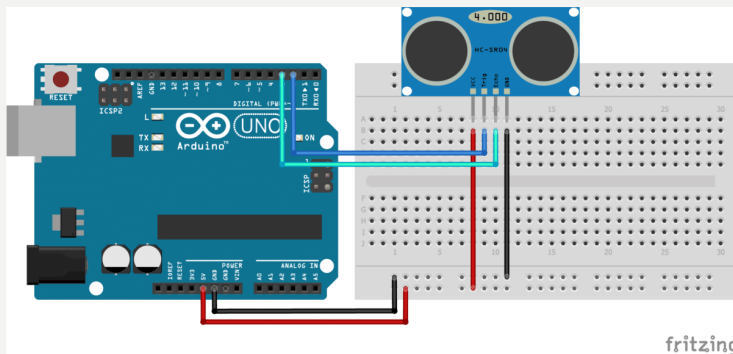
```
void loop() {  
  //  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(5);
```

```
  // Generisanje Trig impulsa  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);
```

```
  // Očitavanje trajanja impulse na echoPin upotrebom pulseIn() u microsec  
  duration = pulseIn(echoPin, HIGH);  
  // Izračunavanje distance  
  distance= duration*0.034/2;
```

```
  // Štampanje distance na serijskom monitoru (Ctrl+Shift+M):  
  Serial.print("Distance = ");  
  Serial.print(distance);  
  Serial.println(" cm");
```

```
  delay(50);  
}
```





# PRIMJER – NEWPING LIBRARY

NewPing (autor Tim Eckel) može se koristiti za mnoge ultrasonic senzore.

Kod je mnogo kraći nego u predhodnom primjeru.

Osim toga, NewPing biblioteka ima i drugih korisnih odobina.

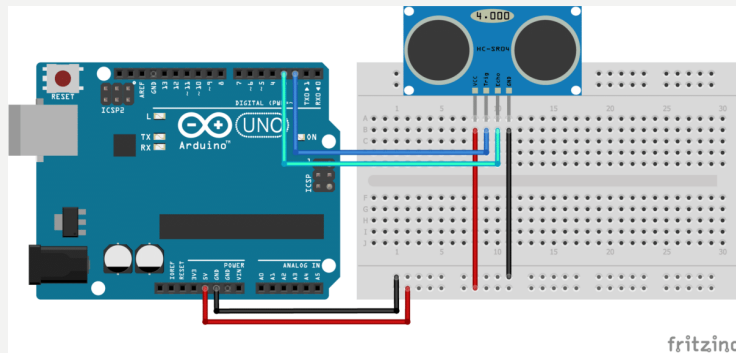
Omogućuje postavljanje maksimalnog dometa, te neće čekati čitavu sekundu ako ne primi eho.

Ima ugrađen median filter.

Može se koristiti `distance = sonar.ping_cm()` ili  
`distance = sonar.ping_in()`

Koji vraćaju mjereno rastojanje u cm ili inčima.

Upotrebom ovih funkcija nije potrebno izračunavanje kao u predhodnom primjeru.



```
// Uključivanje biblioteke
#include <NewPing.h>
```

```
// Definisanje pinova i maksimalnog rastojanja
#define trigPin 2
#define echoPin 3
#define MAX_DISTANCE 350 // Maksimalno rastojanje (u centimetrima). Maksimalan domet --
// senzora je između 400-500cm.
```

```
// NewPing setup
NewPing sonar(trigPin, echoPin, MAX_DISTANCE);
void setup() {
  Serial.begin(9600); // Inicijalizacija serijske komunikacije.
}
```

```
void loop() {
  delay(50); // Čekanje 50ms između dva ping-a (oko 20 ping/sec). 29ms je najkraće vrijeme
```

```
  duration = sonar.ping();
  distance = (duration / 2) * 0.0343;
```

```
  Serial.print("Distance = ");
  Serial.print(distance); // Rastojanje će biti 0 kada je objekat izvan maksimalnog dometa.
  Serial.println(" cm");
}
```

# KAKO KORISTITI MEDIAN DIGITALNI FILTER?

Jedna od važnijih novina NewPing biblioteke je realizacija median filtera.

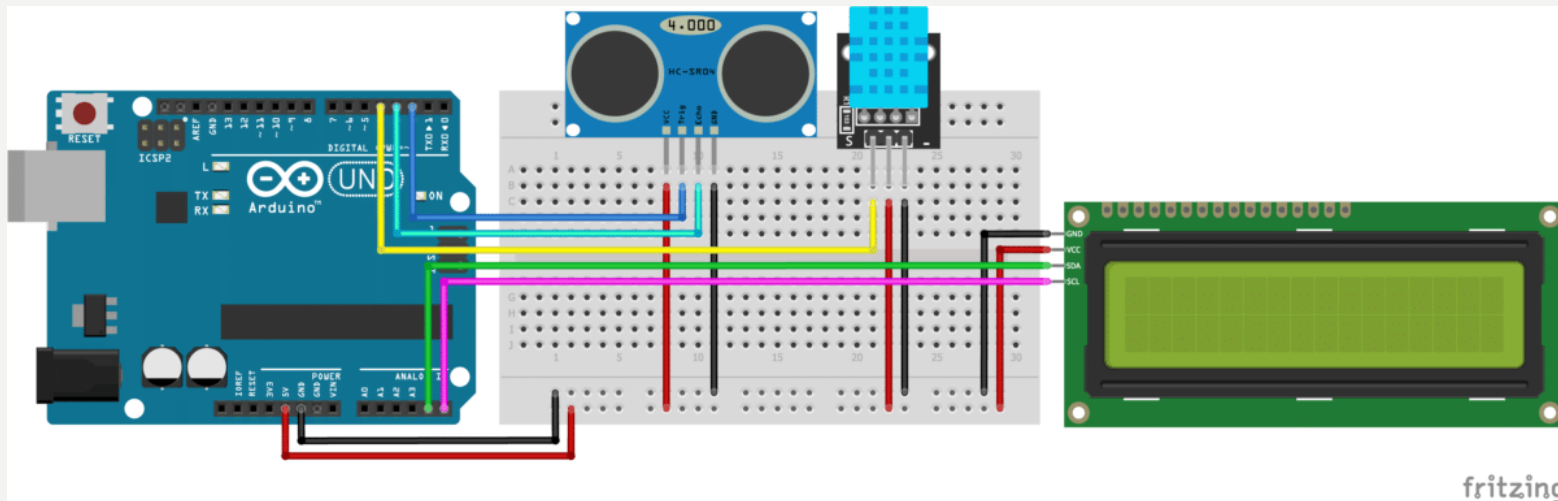
Ovaj filter značajno poboljšava tačnost mjerenja rastojanja.

ping\_median() funkcija uzima puno mjerenja trajanja, tako da se kao zaključna uzima srednja vrijednost.

Najčešće se uzima 5 čitanja ali može se zadati i drugi broj.

```
int iterations = 5;  
duration = sonar.ping_median(iterations);
```

# PRIMJER - HC-SR04 SA DHT11 I I2C LCD



U ovom primjeru koisti se **Adafruit DHT Humidity & Temperature Sensor** biblioteka koja se može preuzeti sa [ovog linka \(GitHub\)](#).

Biblioteka DHT senzora radi jedino ako se ima instalirana **Adafruit\_Sensor** biblioteka, ona se može preuzeti sa [ovog linka \(GitHub\)](#).

Biblioteka **LiquidCrystal\_I2C** može se preuzeti sa [ovog linka \(GitHub\)](#). Uključena je i **Wire.h** biblioteka.

# PRIMJER - HC-SR04 SA DHT11 I I2C LCD

```
#include <Adafruit_Sensor.h> // https://github.com/adafruit/Adafruit_Sensor
#include <DHT.h> // https://github.com/adafruit/DHT-sensor-library
#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // Library for LCD
```

```
// Define Trig pin, Echo pin and DHTPin:
#define trigPin 2
#define echoPin 3
#define DHTPin 4
```

```
// Define SDA and SCL pin from LCD:
#define SDAPin A4 // Data pin
#define SCLPin A5 // Clock pin
```

```
// Connect to LCD via i2c, default address 0x27 (A0-A2 not jumpered):
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27,20,4);
```

```
// Define DHT sensor type:
#define DHTType DHT11
```

```
// Define variables:
long duration; int distance; float speedofsound;
```

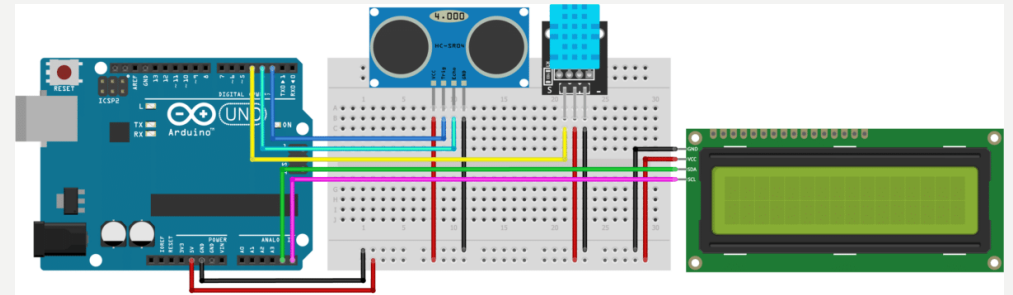
```
// Create a DHT sensor object:
DHT dht = DHT(DHTPin,DHTType);
```

```
void setup() {
  // Define inputs and outputs:
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
```

```
  dht.begin();
```

```
  // Initiate the LCD:
  lcd.init(); lcd.backlight();
```

```
  // Begin Serial communication at a baudrate of 9600:
  Serial.begin(9600);
}
```



fritzing

```
void loop() {
  // Clear the trigPin by setting it LOW:
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);

  // Trigger the sensor by setting the trigPin high for 10 microseconds:
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the echoPin. This returns the duration (length of the pulse) in microseconds:
  duration = pulseIn(echoPin, HIGH);

  // Read the temperature:
  int temperature = dht.readTemperature();

  // Calculate speed of sound in m/s:
  speedofsound = 331.3+(0.606*temperature);

  // Calculate the distance in cm:
  distance = duration*(speedofsound/10000)/2;

  // Print the distance and temperature on the Serial Monitor:
  lcd.setCursor(0,0); lcd.print("T: "); lcd.print(temperature); lcd.print("\xDF" "C ");
  lcd.print("D: "); lcd.print(distance); lcd.print("cm ");
  lcd.setCursor(0,1); lcd.print("Spd: "); lcd.print(speedofsound); lcd.print("m/s ");
  delay(100);
}
```

# PRIMJER ZA VJEŽBU

Senzor pokreta detektuje pokret.

Dok je pokret detektovan uključiti LED i izmjeriti rastojanje od ultrasoničnog senzora.

Za poboljšanje tačnosti mjerenja upotrijebiti sensor temperature i vlage.

Ukoliko je rastojanje manje od 30cm, oglasiti se prigodnom svirkom sa buzzer-a.

Na LCD-u ispisati rastojanje objekta.

Na serijskom monitoru ispisati: Detertovan pokret. Rastojanje: u cm.

Dok pokret nije detektovan na LCD ispisati nema pokreta.

**(3-2 boda)**

# PRIMJER ZA VJEŽBU

Radovi na putu. Suženje puta na jednu kolovoznu traku. Kontrola prolaska.

Dozvolu i zabranu prolaska signalizirati pomoću dvije LED, crvena za zabranu prolaska, zelena za dozvolu prolaska, na obje prilazne strane.

Neka na startu nema automobila ni na jednoj stani. U toj situaciji na obje prilazne strane uključiti crvenu LED.

Kada se na jednoj strani senzorom pokreta ili ultrasoničnim senzorom detektuje nailazak automobila, na toj strani uključiti zelenu LED, a na suprotnoj crvenu LED.

To stanje zadržati dok ne protekne definisano vrijeme (ne koristiti delay!!). Ovo vrijeme zadavati putem serijskog monitora a predefinisati na 15sek.

Po isteku vremena na obje strane uključiti crvenu LED ili ako iz suprotnog smjera već čeka vozilo, invertovati, odnosno sada za taj smjer uključiti zelenu LED a na drugoj strani crvenu. To stanje takođe zadržati isto definisano vrijeme.

**(5-4-3 boda)**

# PRIMJER ZA VJEŽBU

Naplatni punkt na autoputu.

Senzor pokreta raguje dok prilazi vozilo i omogućuje uključenje rasvjete ispred naplatnog šaltera. Rasvjetu predstaviti sa dvije istobojne LED. Rasvjetu uključivati samo ako je mračno, odnosno ako se pomoću fotootpornika detektuje nizak nivo svjetlosti.

Ultrasonični sensor je postavljen iznad vozila na naplatnom punktu i mjeri minimalno rastojanje od vozila koje prolazi. Na osnovu tog podatka vrši klasifikaciju vozila na putničke automobile i teretna vozila (autobuse).

Rasvjetu držati uključenu sve dok vozilo ne prođe ispod ultrazvučnog senzora ili ne protekne maksimalno vrijeme predviđeno za prolazak (ne koristiti delay!!). Maximalno vrijeme zadavati putem serijskog monitora (predefinisati 60sek).

Informaciju o broju putničkih i teretnih vozila koja su prošla od početka nadzora, ispisati na LCD-u i serijskom monitoru.

**(6-5-4 boda)**