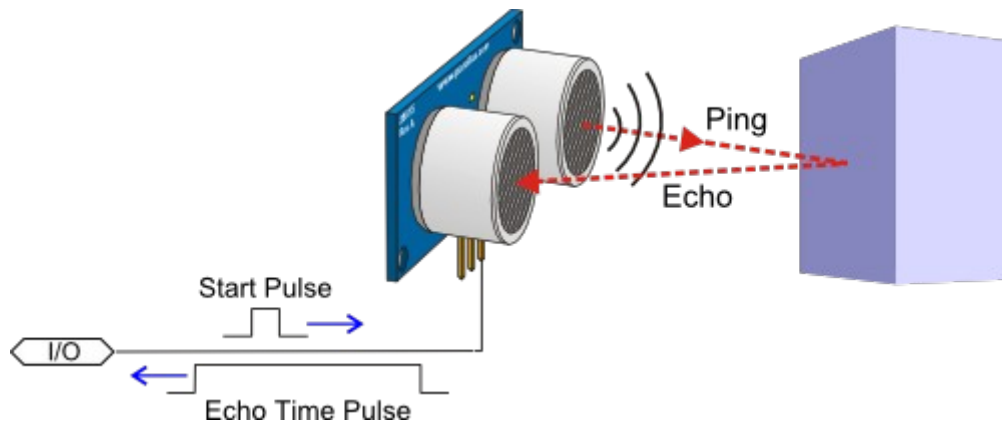


# Αισθητήρας Υπερήχων HC-SR04



Λειτουργία αισθητήρα υπερήχων HC-SR04

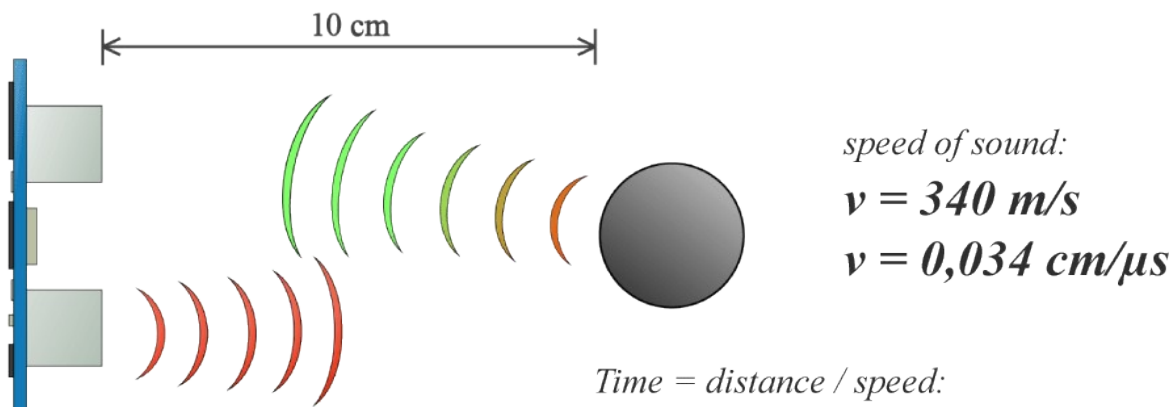
Ο αισθητήρας υπερήχων μπορεί να:

- Ανιχνεύει κοντινά αντικείμενα μέσα σε μία κλίμακα από 2 cm έως 3 μέτρα
- Μετρά αποστάσεις χρησιμοποιώντας ήχο υψηλής συχνότητας (40 kHz)

Πρόκειται για έναν από τους πιο διαδεδομένους και εύχρηστους αισθητήρες απόστασης.

Ο αισθητήρας Απόστασης Υπερήχων HC-SR04 εκπέμπει ένα παλμό υπερήχων στα 40.000 Hz (40kHz) που ταξιδεύει μέσω του αέρα και εάν υπάρχει αντικείμενο ή εμπόδιο στη διαδρομή του, θα επιστρέψει μετά την ανάκλαση στον δέκτη.

Για τον υπολογισμό της απόστασης θα πρέπει να λάβουμε υπόψη μας ότι η ταχύτητα του ήχου είναι  $v=340\text{m/sec}=0,034\text{cm}/\mu\text{sec}$  και ότι ο χρόνος (σε  $\mu\text{sec}$ ) που μας δίνει η ακίδα Echo είναι ο συνολικός χρόνος που ταξίδεψε το κύμα. Επομένως η απόσταση  $s$  (σε cm) του εμποδίου από τον αισθητήρα προκύπτει από τον τύπο:  $s(\text{cm})=t(\mu\text{sec})\cdot 0,034(\text{cm}/\mu\text{sec})/2$ .



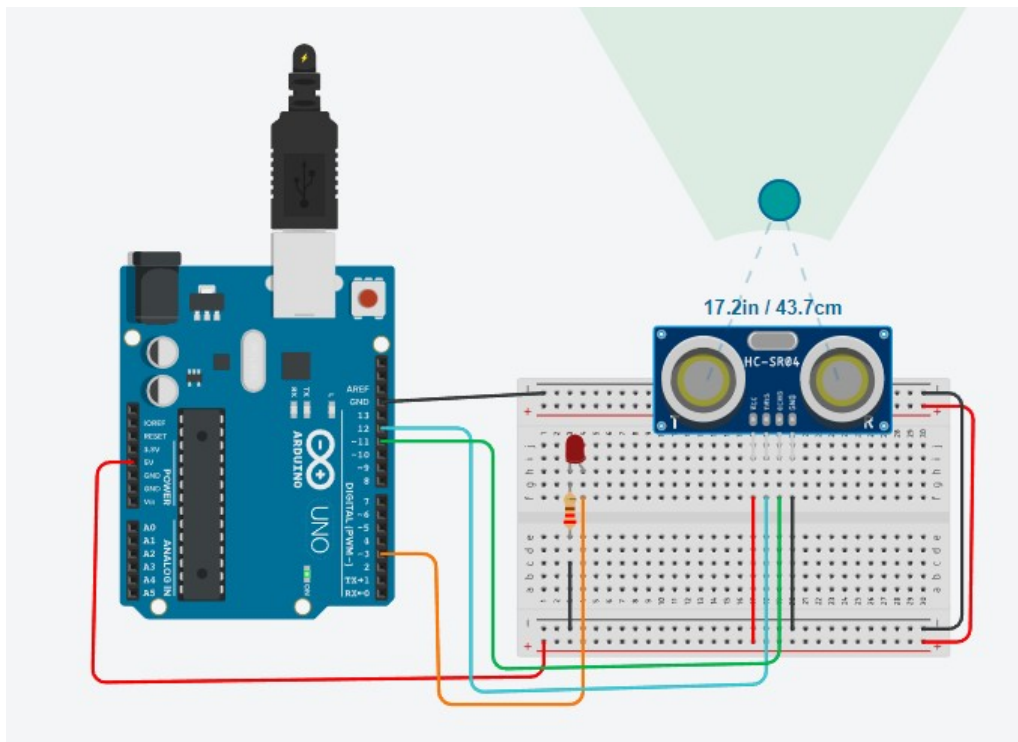
*Time = distance / speed:*

$$t = s / v = 10 / 0,034 = 294 \mu s$$

*Distance:*

$$s = t \cdot 0,034 / 2$$

Παράδειγμα εφαρμογής μέτρησης απόστασης με τον αισθητήρα υπερήχων HC-SR04 στο tinkercad



## Ο κώδικας:

// C++ code

```
const int trigpin=12;  
const int echopin=11;  
const int redled=3;
```

```
long duration;  
int distance;
```

```
void setup(){  
  pinMode(trigpin,OUTPUT);  
  pinMode(echopin,INPUT);  
  pinMode(redled,OUTPUT);
```

```
  Serial.begin(9600);  
}
```

```
void loop(){  
  digitalWrite(trigpin,LOW);  
  delayMicroseconds(2);  
  digitalWrite(trigpin,HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigpin,LOW);  
  duration=pulseIn(echopin,HIGH);
```

```

distance=duration*0.034/2;

if (distance<15){
  digitalWrite(redled,HIGH);
  Serial.print("Distance=");
  Serial.print(distance);
  Serial.println("cm");
}
else{
  digitalWrite(redled,LOW);
  Serial.print("Distance=");
  Serial.print(distance);
  Serial.println("cm");
}
}

```

Με το πρόγραμμα αυτό το led ενεργοποιείται όταν η απόσταση είναι μικρότερη των 15 εκατοστών. Στο serial monitor μπορούμε να παρακολουθούμε την απόσταση αυτή.

Παρατηρείστε ότι κάτω από 2 εκατοστά οι ενδείξεις είναι λανθασμένες καθώς δεν μπορεί κάτω από αυτή την απόσταση δεν ανταποκρίνεται ο αισθητήρας.

Για την ενεργοποίηση της βαλβίδας της εφαρμογής μας αρκεί να συνδέσουμε ένα relay. Στο παράδειγμα που ακολουθεί είναι συνδεδεμένο στο Pin 7 και το relay είναι Low triggered

```

// C++ code
const int trigpin=12;
const int echopin=11;
const int redled=3;
const int relaypin=7;

long duration;
int distance;

void setup(){
  pinMode(trigpin,OUTPUT);
  pinMode(echopin,INPUT);
  pinMode(redled,OUTPUT);
  pinMode(relaypin,OUTPUT);
  Serial.begin(9600);
}
void loop(){
  digitalWrite(trigpin,LOW);
  delayMicroseconds(2);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  distance=duration*0.034/2;
}

```

```
if (distance<15){
  digitalWrite(redled,HIGH);
  digitalWrite(relaypin,LOW);
  delay(10);
}
else{
  digitalWrite(redled,LOW);
  digitalWrite(relaypin,HIGH);

}
Serial.print("Distance=");
Serial.print(distance);
Serial.println("cm");
}
```

Μπορούμε να κρατάμε ενεργοποιημένη την βαλβίδα για προκαθορισμένο χρόνο αν προσθέσουμε στον κώδικα μας κατάλληλες εντολές delay().

Σημείωση : Ο συγκεκριμένος αισθητήρας παρουσιάζει προβλημάτια όταν το εμπόδιο είναι υπό γωνία .