# SENSORE AD ULTRASUONI HC-SR04 – ARDUINO



Misurare la distanza con un sensore ad ultrasuoni hc-sr04 può essere utile in molte occasioni, ad esempio in un sistema anti-intrusione che deve far suonare un allarme o un sistema robotizzato che deve evitare ostacoli. In questo tutorial viene utilizzato il sensore ad ultrasuoni HC-SR04, dispositivo economico e con un buon range operativo. I sensori ad ultrasuoni non forniscono direttamente la misura della distanza dell’oggetto più vicino, ma misurano il tempo impiegato da un segnale sonoro a raggiungere l’oggetto e ritornare al sensore. L’impulso ad ultrasuoni inviato dal HC-SR04 è di circa 40KHz il tempo viene misurato in microsecondi, la tensione di funzionamento è di 5V, quindi potremo alimentarlo direttamente utilizzando Arduino.

# HC-SR04

Il sensore HC-SR04 dispone di 4 pin: Vcc (+5V), Trigger, Echo, GND. Si invia un impulso alto sul pin Trigger per almeno 10 microsecondi, a questo punto il sensore invierà il ping sonoro e aspetterà il ritorno delle onde riflesse, il sensore risponderà sul pin Echo con un impulso alto della durata corrispondente a quella di viaggio delle onde sonore, dopo 38 millisecondi si considera che non sia stato incontrato alcun ostacolo. Per sicurezza si aspettano in genere 50-60 millisec per far si che non vi siano interferenze con la misura successiva. Per maggiori informazioni consultate il [datasheet](http://elecfreaks.com/store/download/HC-SR04.pdf%22%20%5Co%20%22HC-SR04%22%20%5Ct%20%22_blank).



Per convertire l’intervallo di tempo misurato in una lunghezza, bisogna ricordare che la velocità del suono è di 331,5 m/s a 0 °C e di 343,4 m/s a 20 °C ed in generale varia secondo la relazione v = 331,4 + 0,62 T dove la temperatura T è misurata in °C. Per la realizzazione del nostro metro elettronico assumiamo di lavorare ad una temperatura ambiente di 20 °C e quindi la velocità del suono sarà di 343 m/s (approssimiamo) che vuol dire anche 0,0343 cm/microsecondi.

Quindi, ricordando che v=s/t (v: velocità, s: spazio, t: tempo) allora lo spazio percorso sarà:

s = v\*t da cui s = 0,0343 \*t

però, per calcolare lo spazio percorso, bisogna tener conto che il suono percorre due volte la distanza da misurare (giunge sull’oggetto e ritorna indietro al sensore) quindi il valore di t ottenuto deve essere diviso per 2. La formula corretta per la misura dello spazio percorso è:

s = 0,0343 \* t/2

eseguendo la divisione di 0,0343/2 possiamo scrivere:

s = 0,01715 \* t oppure: s = t/58,31

approssimando s = t/58 formula più semplice da ricordare.

# COME CONNETTERE IL SENSORE AD ARDUINO:

 

Per connettere il sensore HC-SR04 ad arduino basta collegare i 2 pin di alimentazione rispettivamente a Vcc e Gnd e gli altri 2 pin (Trigger e Echo) a 2 ingressi digitali.

Ecco lo schema di collegamento:



Scketch con monitor Seriale:

|  |  |
| --- | --- |
| 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849 | //HC RS04 Sensore ultrasuoni//Giuseppe Caccavaleconst int triggerPort = 9;const int echoPort = 10;const int led = 13; void setup() { pinMode(triggerPort, OUTPUT);pinMode(echoPort, INPUT);pinMode(led, OUTPUT);Serial.begin(9600);Serial.print( "Sensore Ultrasuoni: ");} void loop() { //porta bassa l'uscita del triggerdigitalWrite( triggerPort, LOW );//invia un impulso di 10microsec su triggerdigitalWrite( triggerPort, HIGH );delayMicroseconds( 10 );digitalWrite( triggerPort, LOW ); long durata = pulseIn( echoPort, HIGH ); long distanza = 0.034 \* durata / 2; Serial.print("distanza: "); //dopo 38ms è fuori dalla portata del sensoreif( durata > 38000 ){Serial.println("Fuori portata   ");}else{ Serial.print(distanza); Serial.println(" cm     ");} if(distanza < 10){ digitalWrite(led, HIGH);}else{ digitalWrite(led, LOW);} //Aspetta 1000 microsecondidelay(1000);} |

Scketch con display lcd:

|  |  |
| --- | --- |
| 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455 | //HC RS04 Sensore ultrasuoni//Giuseppe Caccavale#include <LiquidCrystal.h>const int triggerPort = 9;const int echoPort = 10;const int led = 13; LiquidCrystal lcd(2, 3, 4, 5, 6, 7); void setup() { pinMode(triggerPort, OUTPUT);pinMode(echoPort, INPUT);pinMode(led, OUTPUT);lcd.begin(16, 2);lcd.setCursor(0, 0);lcd.print( "Ultrasuoni ");} void loop() { //porta bassa l'uscita del triggerdigitalWrite( triggerPort, LOW );//invia un impulso di 10microsec su triggerdigitalWrite( triggerPort, HIGH );delayMicroseconds( 10 );digitalWrite( triggerPort, LOW ); long durata = pulseIn( echoPort, HIGH ); long distanza = 0.034 \* durata / 2; lcd.setCursor(0, 1);lcd.print("dist.: "); //dopo 38ms è fuori dalla portata del sensoreif( durata > 38000 ){lcd.setCursor(0, 1); lcd.println("Fuori portata   ");}else{ lcd.print(distanza); lcd.println(" cm     ");} if(distanza < 10){ digitalWrite(led, HIGH);}else{ digitalWrite(led, LOW);} //Aspetta 1000 microsecondidelay(1000);} |

**AGGIORNAMENTO**

Di recente ho trovato per il web una libreria in grado di migliorare la lettura e la sensibilità di questo sensore. Potete scaricare la libreria [**QUI**](https://www.dropbox.com/s/s47utngoflgahjn/NewPing.zip?dl=0). Ricordate di copiare la libreria in   **\Documents\Arduino\libraries\…** oppure in **C:\ProgramFiles\Arduino\libraries\…**

Posto qui un esempio, i collegamenti sono uguali a quelli dell’esempio con monitor seriale:

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627 | //HC RS04 Sensore ultrasuoni//Giuseppe Caccavale #include <NewPing.h>#define TRIGGER\_PIN 2 // Arduino pin tied to trigger pin on the ultrasonic sensor.#define ECHO\_PIN 3 // Arduino pin tied to echo pin on the ultrasonic sensor.#define MAX\_DISTANCE 400 // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at 400-500cm. NewPing sonar(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANCE); int SetDistance = 0;int ValueDist = 0; void setup() { Serial.begin(115200);} void loop() { delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay between pings. unsigned int uS = sonar.ping(); // Send ping, get ping time in microseconds (uS). ValueDist = uS / US\_ROUNDTRIP\_CM;  Serial.print("Ping: "); Serial.print(ValueDist); // Convert ping time to distance in cm and print result (0 = outside set distance range) Serial.println("cm"); } |