

## SENSOR ULTRASÓNICO

Es un pequeño radar que permite medir distancia hasta obstáculos. Emite un pulso de sonido a una alta frecuencia no audible por el oído humano y mide el tiempo que tarda el sonido en llegar al obstáculo, rebotar y volver al sensor. Como conocemos la velocidad del sonido en el aire (343,2 m/s) podemos calcular la distancia con  $e = v * t$ . (espacio = velocidad por tiempo).

El rango de medida de los sensores suele ir desde algunos centímetros a varios metros.

Los más comunes, como el HC-SR04, tienen 4 pines. Dos de ellos son para alimentación (Vcc y GND) y los otros dos pines son el *Trigger* y el *Echo*.



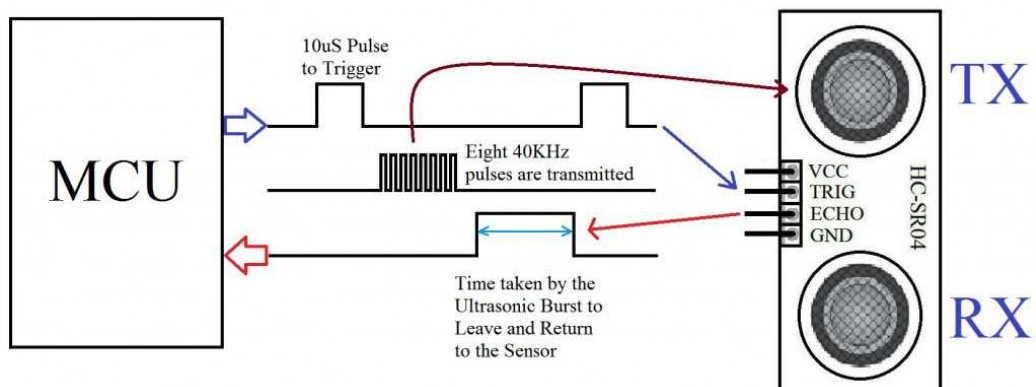
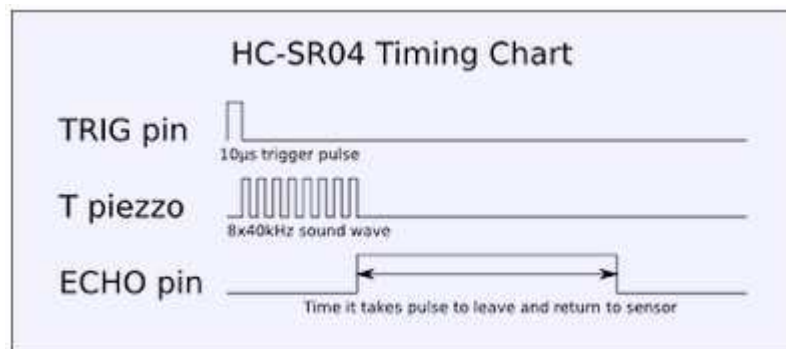
### Funcionamiento

Cuando se le proporciona un nivel alto (HIGH, 5V)

de al menos 10  $\mu$ s al pin "Trig", el módulo ultrasónico emite una serie de 8 pulsos a una alta frecuencia (40 KHz) y a continuación pone su salida "Echo" a HIGH. Si la señal encuentra algún obstáculo, se reflejará y será



captada por el módulo sensor. En el momento de captar la señal de rebote, el módulo cambiará su salida "Echo" de HIGH a LOW. Por tanto, en la salida "Echo" se ha mantenido un pulso de nivel alto (HIGH) con una duración igual al tiempo empleado por la señal en ir desde el sensor al obstáculo y volver. El ancho del pulso varía aproximadamente entre 150  $\mu$ s y 25 ms. Si no encuentra obstáculo, el ancho del pulso de eco será de alrededor de 38 ms.



La duración del pulso de eco la mediremos con Arduino con la instrucción **pulseIn()** . La función pulseIn (pin, nivel) devuelve el tiempo en  $\mu\text{s}$  que dura un pulso en el pin indicado. Su Timeout por defecto es 1 segundo: si al cabo de este tiempo no ha terminado el pulso devuelve 0.

Un ejemplo de código sería el siguiente. Calcula la distancia a un obstáculo y la envía por el puerto serie.

```

/* Sketch medidos ultrasónico*/

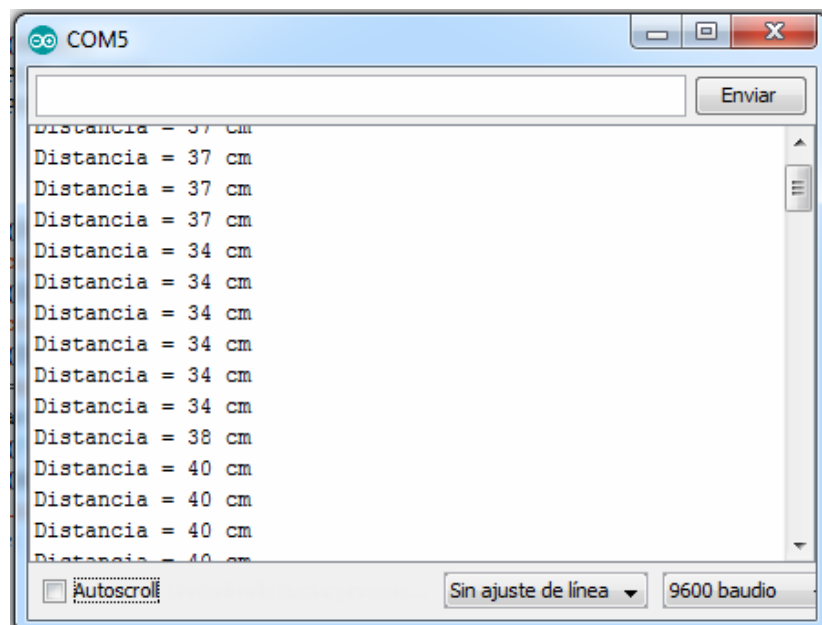
#define TRIGPIN 11
#define ECHOPIN 12

void setup() {
  Serial.begin(9600);
  pinMode(TRIGPIN, OUTPUT);
  pinMode(ECHOPIN, INPUT);
}

void loop() {
  digitalWrite(TRIGPIN, LOW); //Esto se hace para estabilizar el sensor
  delayMicroseconds(2);
  digitalWrite(TRIGPIN, HIGH); //iniciamos el pulso
  delayMicroseconds(10); //esperamos 10 microsegundos
  digitalWrite(TRIGPIN, LOW); //Terminamos el pulso
  float tiempo=pulseIn(ECHOPIN, HIGH); //medimos el tiempo a HIGH de pin ECHO
  int distancia=int(tiempo*0.03432/2); //Calculamos la distancia en centímetros
  Serial.print("Distancia = ");
  Serial.print(distancia);
  Serial.println(" cm");
  delay (1000); //Esperamos un segundo antes de la siguiente medición
}

```

En el monitor del puerto serie se obtendrá una serie de mediciones como las siguientes:



## LIBRERÍA ULTRASONIC

Otra opción es utilizar una librería, como la Ultrasonic.h

El siguiente código imprime en el monitor serie la distancia en pantalla cada medio segundo:

```
// Ultrasonic - Library for HC-SR04 Ultrasonic Ranging Module.

#include <Ultrasonic.h>

Ultrasonic ultrasonic(9,8); // (Trig PIN,Echo PIN)

void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.print(ultrasonic.Ranging(CM)); // CM or INC
  Serial.println(" cm" );
  delay(500);
}
```

El siguiente código imprime en el monitor serie el tiempo invertido por el sonido en llegar al obstáculo y volver al sensor en milisegundos.

```
// Ultrasonic - Library for HC-SR04 Ultrasonic Ranging Module.

// more info at www.ardublog.com

#include <Ultrasonic.h>

Ultrasonic ultrasonic(9,8); // (Trig PIN,Echo PIN)

void setup() {
  Serial.begin(9600);
}

void loop()
{
  Serial.print(ultrasonic.Timing());
  Serial.println(" ms" ); // milliseconds
  delay(500);
}
```

La función `ultrasonic.Timing()` devuelve el tiempo en milisegundos que tarda el sonido en ir y volver.

La función `ultrasonic.Ranging(CM)` devuelve la distancia al obstáculo en cm

Con esta librería, el tiempo de espera por defecto a que termine el pulso de eco está en 3 ms, (es decir, si a los 3 ms el sensor no ha detectado el retorno de la señal, da por terminado el pulso de HIGH, dando como distancia la correspondiente a un pulso de 3 ms, es decir, sobre 51 cm). Por este motivo, la distancia máxima que puede medir está en unos 51 cm. Esto se ha diseñado así porque si no hay obstáculos cercanos el programa se para 1 segundo (1 s es el Timeout por defecto de la función `pulseIn()` en que se basa esta librería) esperando el final del pulso, lo que puede afectar a la captura de otros eventos.

Si se desea aumentar el rango de medida, se puede cambiar el renglón:

```
Ultrasonic ultrasonic(9,8); // (Trig PIN, Echo PIN)
```

Por el renglón:

```
Ultrasonic ultrasonic(9,8,3000); // (Trig PIN, Echo PIN, Timeout en µs)
```

En este caso el tiempo de espera (Timeout) es 3000 (son microsegundos, por lo que son 3 ms). Podemos cambiar este valor para cambiar la distancia que se puede medir.

La relación entre la distancia que se puede medir y el parámetro Timeout es:

$$\text{Timeout en microsegundos} = \text{Distancia en cm} * 58$$

Por ejemplo, si queremos medir distancias de hasta 500 cm, ponemos un Timeout como mínimo de 29000 µs.

Información en:

<https://github.com/JRodrigoTech/Ultrasonic-HC-SR04/wiki/How-to-change-the-default-tiemout%3F>