

# Tutorial\_07.- Arduino\_Sensor de distancia por ultrasonidos

## Contenido

Sensores de proximidad por ultrasonidos .....	3
Contenido teórico .....	3
Características .....	3
Pines .....	4
Funcionamiento .....	4
Funciones de Arduino.....	5
Proyecto07_01: Lectura de un sensor de ultrasonidos HC-SR04.....	6
Ejercicios.....	8
Problemas con los sensores de ultrasonidos .....	9

### Material necesario

- Placa Arduino
- Cable micro USB
- Placa Protoboard
- Componentes:

- Sensores: HC-SR04

### Funciones propias de Arduino

- pulseIN()

#### **IMPORTANTE:**

Comprobaremos que nuestra placa **Arduino** está **desconectada** y sin energía, puesto de no ser así podría dañarse tanto la placa, como el equipo. Una vez hemos realizado esta comprobación, pasaremos a realizar el montaje.

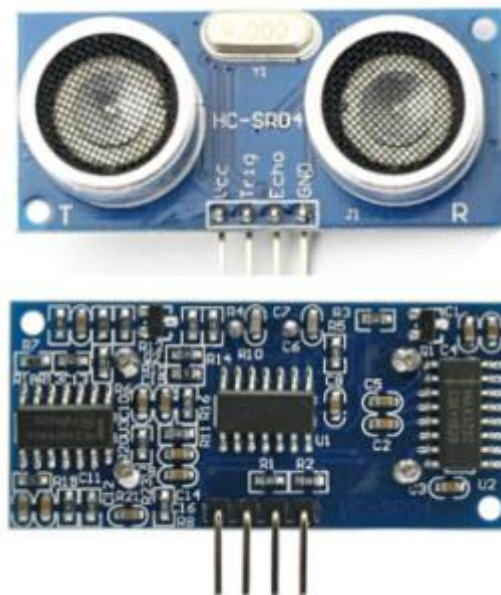
# Sensores de proximidad por ultrasonidos

## Contenido teórico

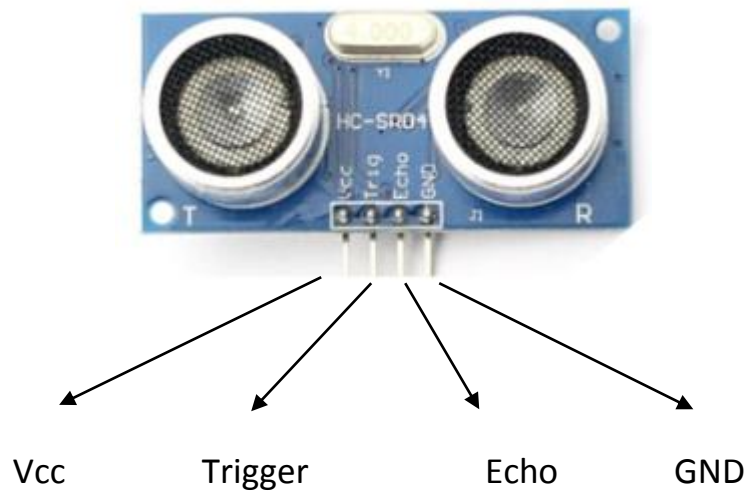
El sensor ultrasónico HC-SR04 determina la distancia a un objeto midiendo el tiempo que tarda en ir y volver una onda ultrasónica. A diferencia del sensor sharp de infrarrojos, no se ve influenciado por la luz ambiente. Este modelo de sensor incorpora tanto el transmisor como el receptor de ultrasonido.

## Características

- Power Supply :+5V DC
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm
- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm



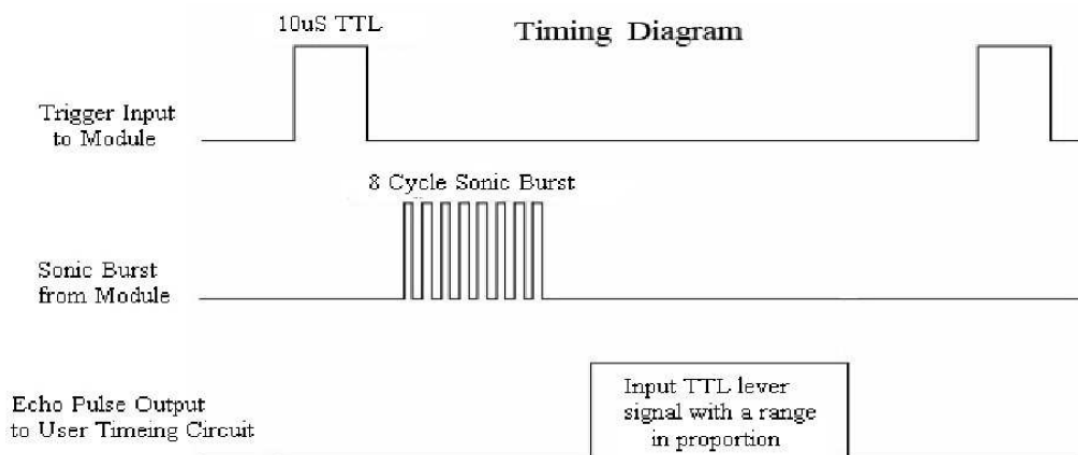
Pines



## Funcionamiento

Atendiendo a las explicaciones de la hoja de características, el terminal “Trigger” debe mantenerse a nivel alto durante al menos 10  $\mu$ s. Entonces, de forma automática, el módulo envía 8 ciclos de una señal de 40KHz y justo después establece la salida “Echo” a nivel alto. “Echo” se mantiene a nivel alto hasta que recibe la señal rebotada, si choca contra un objeto, momento en que va a nivel bajo.

De este modo, el tiempo que la ráfaga tarda desde que sale del sensor hasta que vuelve, será proporcional al tiempo que “Echo” se mantiene con valor alto.



Por otra parte las especificaciones del módulo sensor recomienda no repetir el disparo de la ráfaga hasta que no hayan transcurrido 60ms desde el anterior. Si no se respeta este intervalo, puede suceder que reciba la onda de retorno de un disparo anterior. (Ver siguiente apartado sobre Problemas con los sensores ultrasónicos)

## Funciones de Arduino

### pulseIn()

#### Description

Reads a pulse (from HIGH to LOW or from LOW to HIGH) on a pin. For example, if value is set to HIGH, pulseIn() waits for the pin to go HIGH, starts timing, then waits for the pin to go LOW and stops timing. Returns the length of the pulse in microseconds or 0 if no complete pulse was received within the timeout.

The timing of this function has been determined empirically and will probably show errors in shorter pulses. Works on pulses from 10 microseconds to 3 minutes in length. Please also note that if the pin is already high when the function is called, it will wait for the pin to go LOW and then HIGH before it starts counting.

#### Syntax

1. pulseIn(pin,value)
2. pulseIn(pin, value, timeout)

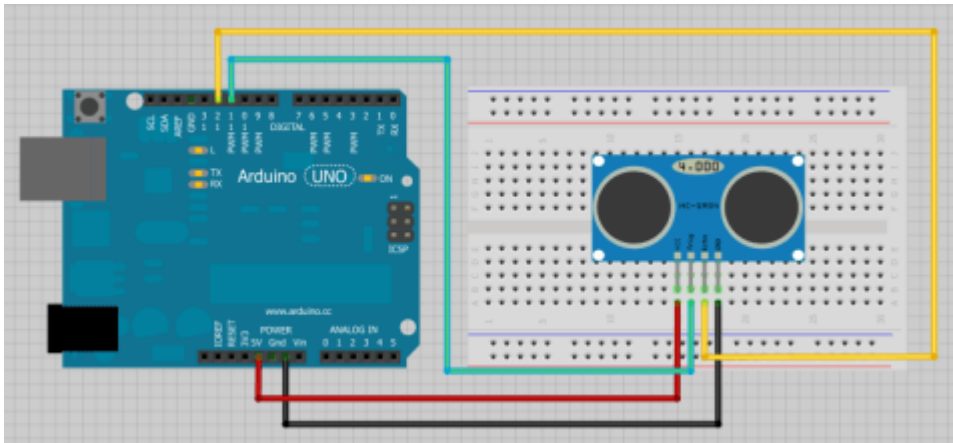
#### Parameters

- **pin**: the number of the pin on which you want to read the pulse. Tipo INT
- **value**: type of pulse to read: either HIGH or LOW.
- **timeout** (optional): the number of microseconds to wait for the pulse to be completed:
  - The function returns 0 if no complete pulse was received within the timeout.
  - Default is one second.

#### Returns

The length of the pulse (in microseconds) or 0 if no pulse is completed before the timeout (unsigned long).

## Proyecto07\_01: Lectura de un sensor de ultrasonidos HC-SR04



### Código

```
/* Mirian Cifredo - University of Cádiz
 * 24/06/2015
 */

/*Variables*/
float distance;
unsigned long time_bounce;

void setup() {
  Serial.begin(9600); // Setup the serial port
  pinMode(9, OUTPUT); // Setup pin9 as an output.(TRIGGER)
                      // It will send the ultrasonic ping

  pinMode(8, INPUT); // Setup pin9 as an input.(ECHO)
                     // It will receive the ultrasonic bounce
}

void loop() {
  /* Keep to LOW the sensor for 5us to ensure a later rising-edge */
  digitalWrite(9, LOW);
  delayMicroseconds(5);

  /* Send a pulse HIGH for 10 us (According to datasheet) */
  digitalWrite(9, HIGH);
  delayMicroseconds(10);
  digitalWrite(9, LOW);

  /* "pulseIn" measures how many takes a signal(in microseconds) goes from
  HIGH (when it receives the bounce) to LOW(when it stops to receive the
  bounce) or LOW to HIGH.
  "pulseIN" works right on pulses from 10 microseconds to 3 minutes in
  length.*/
  time_bounce = pulseIn(8, HIGH);
}
```

```

/* Formula to figure out the distance
 * speed = space /time --> space = speed * time
 * sound speed = 340 m/s --> 0.034 cm/us
 * space --> to travel from the sensor to the object and from de object to
sensor
 * distance to object = space / 2
 */
distance = 0.017 * time_bounce; //Fórmula para calcular la distancia

/*Show in cm the distance measured, by serial monitor */
Serial.println("Distancia ");
Serial.print(distance);
Serial.println(" cm");

/* ??? */
delay(1000);
}

```

La alternativa a pulseIN es hacerlo con interrupciones!!! Y temporizador

## Ejercicios

1. Calcule si la función “pulseIn()” puede darse por válida para el módulo de ultrasonidos utilizado en clase. Según la hoja de características, el sensor es capaz de medir la distancia a un objeto situado a una distancia mínima de 2cm y máxima de 3m.

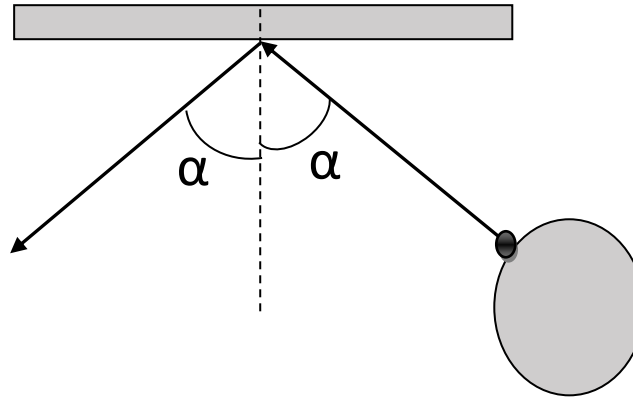
(NOTA: revise el tiempo de duración del pulso que la función es capaz de captar)



## Problemas con los sensores de ultrasonidos

### 1) No recepción de la onda rebotada

La ondas ultrasónicas siguen las propiedades de otras ondas, el ángulo con que el rayo incide en una superficie,  $\alpha$ , es el mismo con que se refleja



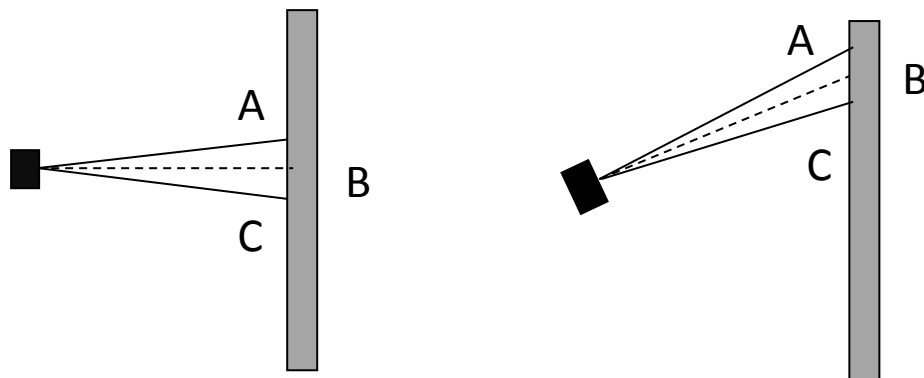
Luego la orientación ideal entre el eje del sensor y el objeto a detectar es  $90^\circ$ , cuanto mas varíe respecto a este valor mayor probabilidad de que no se reciba la señal rebotada.

### 2) Distancia medida errónea

- a) Haz cónico: debe tenerse en cuenta que el haz emitido tiene forma cónica lo que da lugar a que incide en varios puntos del objeto, p.e. A, B y C.

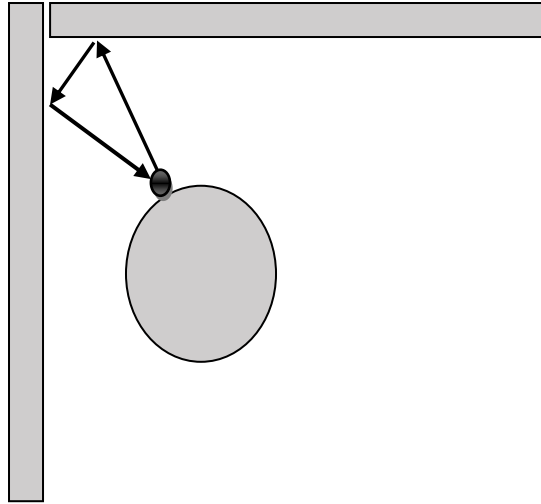
Observando la figura de abajo se aprecia que cuando el sensor es totalmente perpendicular al objeto llegará antes al punto B que al A o al C. En este caso, se tendrá una medición correcta.

Sin embargo, cuando la orientación del sensor respecto al objeto no es perpendicular será el punto C y no B el que primero provoque una señal de rebote. En este caso la medición no será demasiado precisa.



b) Reflexión múltiple

Si la onda emitida por el transductor se refleje varias veces en distintas superficies antes de que regrese al sensor la medición será errónea. Piense que el tiempo transcurrido será mayor que el que realmente tardó en chocar con el objeto.



3) Recepción de la onda rebotada de otro sensor (Crosstalk)

La figura de abajo representa el caso de un robot con más de un sensor ultrasónico de modo que el haz emitido por el sensor1 es recibido en realidad por el sensor2.

