

Práctica 5

Uso del sensor de ultrasonidos para medir distancias

Grado en Ingeniería en Robótica Software

GSyC, Universidad Rey Juan Carlos



(CC) Julio Vega

1. Introducción

En el Tema 5 estudiaremos en profundidad los sensores de nivel y proximidad, entre los cuales destaca el sensor de ultrasonidos. En el mercado existen numerosos modelos de estos sensores, así como utilidades prácticas.

Es un sensor muy empleado en robótica, pues aporta información muy precisa sobre distancia a objetos y su programación es bastante sencilla.

Un sonido se califica como ultrasonido cuando la frecuencia es mayor que la frecuencia audible por el oído humano, que aproximadamente es de 20KHz . El funcionamiento se basa en el ya comentado efecto Doppler, pues la onda emitida por el dispositivo es atenuada por objeto cuya distancia se pretende medir.

Pues bien, la medición de esa atenuación por parte del receptor es la clave de este tipo de sensores. El núcleo del sensor es un material piezoeléctrico. Recordemos que la piezoelectricidad se da cuando se produce una tensión de salida V_{out} debida a una presión ejercida sobre el material, o viceversa; esto es, debido a una excitación eléctrica, el material se deforma y emite una onda mecánica o, como sucede en este caso, emite una onda ultrasónica.

La onda se emite de forma cíclica a alta frecuencia y de corta duración. Esta onda viaja por el medio hasta que es reflejada en algún objeto y, entonces, regresa su *eco*. Mediante un circuito de acondicionamiento se determina el periodo de tiempo transcurrido entre la emisión de onda y la recepción de su eco. La fórmula empleada para ello se muestra en la Ecuación 1.

$$d = \frac{1}{2}v_s t \quad (1)$$

donde:

d : distancia del emisor-receptor al objeto [m]

v_s : velocidad del sonido

t : tiempo transcurrido [s]

2. Sensor de ultrasonidos HC-SR04

Dentro del kit que se ha facilitado en la asignatura, encontramos el sensor de ultrasonidos modelo HC-SR04, como el que vemos en la Figura 1, y cuya hoja de especificaciones, extraída de Sparkfun Electronics¹, se adjunta.

¹<https://sparkfun.com>

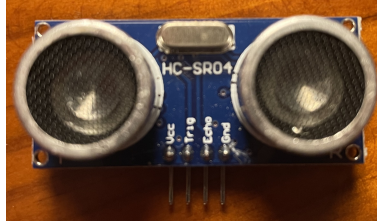


Figura 1: Sensor de ultrasonidos HC-SR04 incluido en el kit

Este modelo es un sensor de ultrasonidos cuya mayor utilidad es medir la distancia de este a cualquier objeto en el que rebota el ultrasonido en modo *reflexión directa*. Para ello, se contabiliza el tiempo que tarda la señal ultrasónica de $40kHz$ del transmisor en rebotar sobre una superficie y ser devuelta al receptor.

Teniendo en cuenta este tiempo y la velocidad a la que viaja el sonido en el aire, se puede obtener la distancia al objeto con una precisión bastante elevada.

2.1. Descripción de los pines del sensor

Este modelo cuenta con cuatro terminales o pines. Cada uno de ellos tiene una función que se detalla a continuación:

- VCC. Pin de alimentación del sensor (5V).
- TRIG. Pin de disparo (salida) que habilita la medición del sensor.
- ECHO. Señal de entrada al sistema.
- GND. Pin negativo de alimentación o toma de tierra.

2.2. Diagrama de tiempos

En la Figura 2, extraída de la hoja de especificaciones, podemos ver el funcionamiento de este modelo.

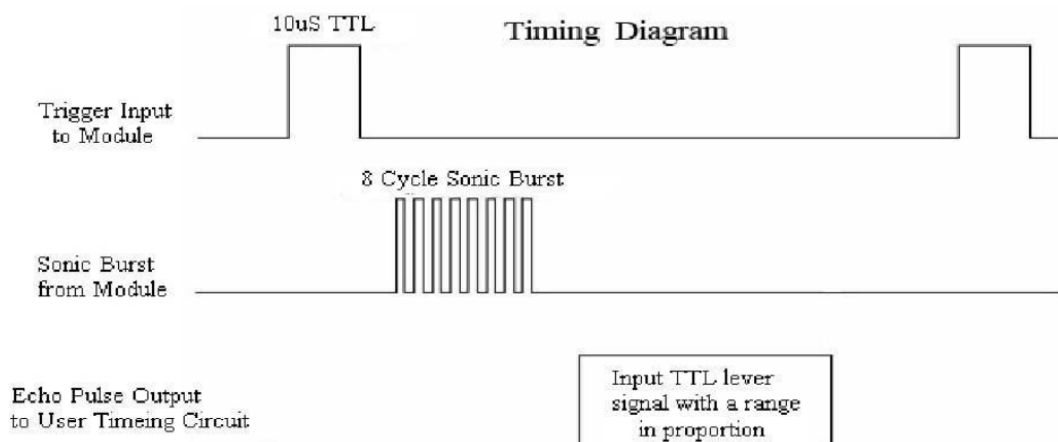


Figura 2: Diagrama de tiempos del modelo HC-SR04

Como vemos, para comenzar con la medición, es necesario aplicar un pulso, una señal en alto, a través del pin *Trig*, con una duración de —al menos— $10\mu s$. A continuación, el sensor envía una serie de 8 pulsos de $40kHz$. Y, tras esto, pone el pin de *Echo* en alto, y permanecerá en ese estado hasta que se reciba el eco de los pulsos de $40kHz$. Es decir, este tiempo es el que se toma en cuenta para estimar la distancia a la que se encuentra el objeto.

2.3. Cálculo de la distancia al objeto

Como todos sabemos, la fórmula que relaciona la distancia con la velocidad es como se refleja en la Ecuación 2.

$$v = \frac{s}{t} \iff d = v \cdot t \quad (2)$$

Considerando que la velocidad del sonido en el aire es de $343 \frac{m}{s}$, y sustituyendo en la Ecuación 2, nos queda la Ecuación 3.

$$d = v \cdot t \implies d = 343 \frac{m}{s} \cdot t \quad (3)$$

En nuestro caso, como estamos usando el sensor en modo *reflexión directa*, la distancia d corresponde a la distancia recorrida por el tren de 8 pulsos desde que sale del módulo, hasta que vuelve al ser rebotada en el objeto. Dicho de otro modo, d es —por tanto— el doble de la distancia real d_r entre el sensor y el objeto. Así, esta d_r nos queda expresada apropiadamente en la Ecuación 4.

$$d = 343 \frac{m}{s} \cdot t \implies d_r = 171,5 \frac{m}{s} \cdot t \quad (4)$$

Por último, transformamos la Ecuación 4 a la Ecuación 5, donde hemos adaptado las unidades a las que estamos manejando en este problema en particular (cm y μs), para que nos resulte más cómodo trabajar con ella.

$$d_r = 171,5 \frac{m}{s} \cdot t = 17150 \frac{cm}{s} \cdot t(\mu s) \cdot 10^{-6} \frac{s}{\mu s} \quad (5)$$

3. Práctica

Una vez —y solo entonces— que hemos leído el *datasheet* y hemos interiorizado la teoría, podemos ponernos manos a la obra y montar el circuito, así como proceder a su programación.

3.1. Conexión del dispositivo

El esquema de conexión se puede observar en la Figura 3, cuyo funcionamiento se resume a continuación:

- VCC a pin de 5V directamente.
- TRIG a un pin GPIO.
- Divisor de voltaje para recibir ECHO (como ya hemos aprendido, e.g. $2,2K\Omega - 1K\Omega$, o también con $470\Omega - 220\Omega$:
 - ECHO a resistencia de $1K\Omega$.
 - La otra resistencia de $2,2K\Omega$ conecta en la misma línea de la otra resistencia hacia *Ground*.
 - Cable desde esa línea (entre las dos resistencias) hacia otro pin GPIO.
- GND a *Ground*.

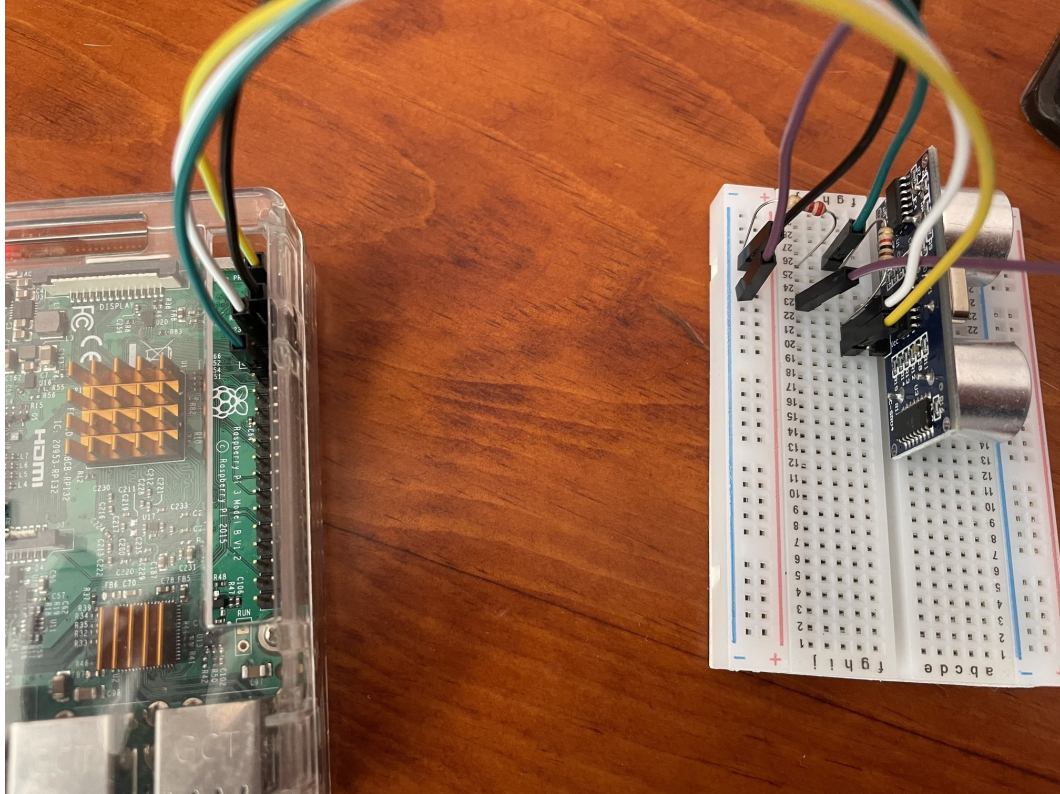


Figura 3: Esquema de conexión del US modelo HC-SR04

3.2. Programación del ultrasonidos

Una vez tengas el circuito conectado, prueba el funcionamiento del ultrasonidos con el código que se facilita en `usInstantaneo.py` y cuyo snippet se muestra a continuación. En este código se toma una única medición del US.

```
#!/usr/bin/python
import RPi.GPIO as GPIO
import time

try:
    GPIO.setmode(GPIO.BCM)

    PIN_TRIGGER = 4
    PIN_ECHO = 17

    GPIO.setup(PIN_TRIGGER, GPIO.OUT)
    GPIO.setup(PIN_ECHO, GPIO.IN)

    GPIO.output(PIN_TRIGGER, GPIO.LOW)

    print "Esperando a que se estabilice el US"
    time.sleep(2)

    GPIO.output(PIN_TRIGGER, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(PIN_TRIGGER, GPIO.LOW)
```

```
while GPIO.input(PIN_ECHO)==0:
    inicioPulso = time.time()
while GPIO.input(PIN_ECHO)==1:
    finPulso = time.time()

duracionPulso = finPulso - inicioPulso
distancia = round(duracionPulso * 17150, 2)
print "Distancia: ", distancia, " cm"

finally:
    GPIO.cleanup()
```

3.3. Ejercicio

Adapta el código anterior para que el interfaz sea más amigable. Para ello, haz uso de tres LEDs que indiquen el acercamiento del objeto según tu criterio:

1. Luz verde. Inicialmente, será el LED que estará encendido, indicando que existe una distancia de seguridad razonable al objeto.
2. Luz amarilla. Si el objeto entra en un tramo intermedio de distancia, que consideremos prudencial, aunque no crítica.
3. Luz roja. Si el objeto entra en una zona peligrosa de distancia; esto es, está excesivamente cerca.

Es muy interesante hacer pruebas experimentales para, por ejemplo, establecer el rango del sensor; e.g. distancia mínima/máxima de detención de objetos.