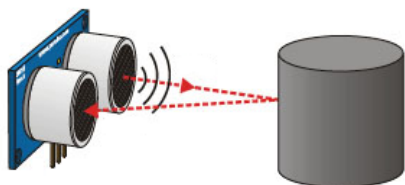


PRÁCTICA 16: Medición de la distancia con sensor de ultrasonidos.

En esta práctica explicaremos como utilizar un **sensor ultrasónico HC-SR04** para con ayuda de **Arduino** medir distancias y, de esta manera, conseguir que nuestros mecanismos sean capaces de detectar obstáculos. El sensor consta de un **terminal emisor** que emite un **ultrasonido** (inaudible para el oído humano por su alta frecuencia) y **espera un determinado tiempo con el terminal receptor la vuelta de la onda sonora**. Si no detecta nada es debido a que no hay ningún obstáculo en el rango de medidas del sensor (este sensor en concreto tiene un rango de distancias sensible entre 3cm y 3m con una precisión de 3mm) mientras que si recibe la respuesta usa el hecho de que el sonido lleva un movimiento rectilíneo uniforme para, a partir del tiempo que pasa desde que emite hasta que recibe la señal, realice de forma sencilla el cálculo de la distancia a la que se encuentra el objeto.

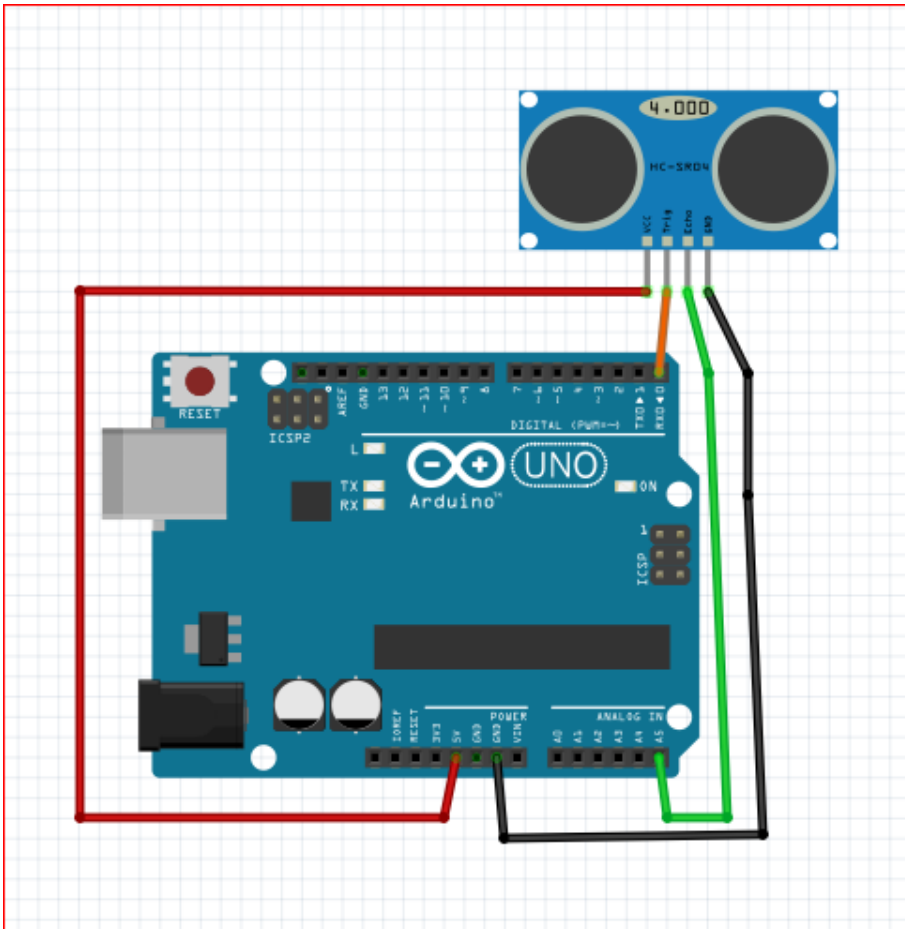


$$\text{Tiempo} = 2 * (\text{Distancia} / \text{Velocidad})$$
$$\text{Distancia} = \text{Tiempo} \cdot \text{Velocidad} / 2$$



Circuito eléctrico (diagrama de conexiones)

La primera patilla irá a **5 V (VCC)** mientras que la última irá a **tierra (GND)**, la patilla del **emisor o trigger** se conectará a un pin digital (en nuestro caso el pin 0) y el **receptor o echo** se se conectará a un pin analógico (en nuestro caso el pin analógico A5)



El siguiente programa es una especie de theremin casero en el que según la distancia de la mano al sensor se ejecutará una nota musical diferente.

The image shows a Scratch script for a distance-based note player. It starts with an 'al presionar' (when pressed) event block, followed by a 'por siempre' (forever) loop. Inside the loop, the first block is 'fijar distancia a valor del sensor Analog5', which sets the 'distancia' variable to the value of the 'Analog5' sensor. This is followed by seven 'si' (if) blocks, each with a range condition and a 'tocar nota' (play note) block. The conditions and notes are as follows:

- Si $distancia < 10$ y $distancia > 0$: tocar nota 48 durante 0.5 pulsos
- Si $distancia < 15$ y $distancia > 9$: tocar nota 50 durante 0.5 pulsos
- Si $distancia < 20$ y $distancia > 14$: tocar nota 52 durante 0.5 pulsos
- Si $distancia < 25$ y $distancia > 19$: tocar nota 53 durante 0.5 pulsos
- Si $distancia < 30$ y $distancia > 24$: tocar nota 55 durante 0.5 pulsos
- Si $distancia < 35$ y $distancia > 29$: tocar nota 57 durante 0.5 pulsos
- Si $distancia < 100$ y $distancia > 34$: tocar nota 59 durante 0.5 pulsos

The script ends with a small arrow icon at the bottom of the loop, indicating it repeats forever.

Sensor HC-SR04

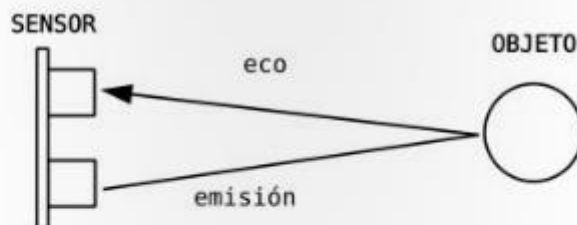
Los sensores de ultrasonidos son elementos que, al incluirlos en un robot, le aportan la posibilidad de orientarse por el espacio, consiguiendo que el robot detecte objetos.

HC-SR04 es un sensor de ultrasonidos porque emite a una frecuencia muy elevada, del orden de 40KHz. Frecuencia que el oído humano no logra percibir. Físicamente dispone de 4 pines: **Vcc** y **GND** que son los de alimentación (+5V y masa) respectivamente, y **Trig** y **Echo** que son los pines de datos. Para realizar una medición que nos aporta una distancia (mide distancias entre un rango de 2cm a 450cm), se debe aplicar un pulso en Trig y luego leer el "rebote" en Echo:



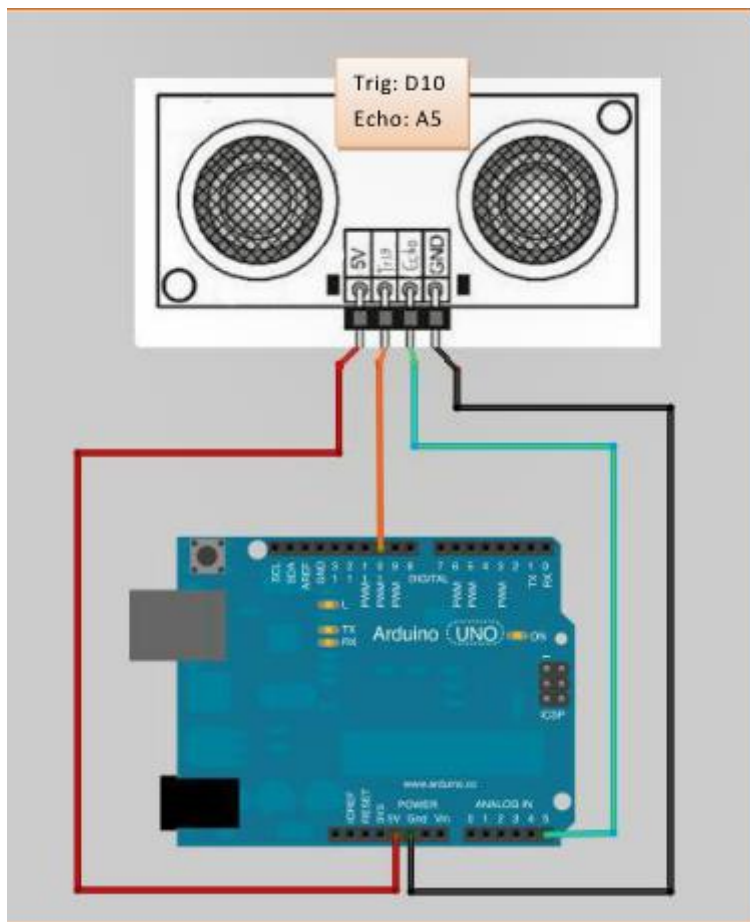
Sensor HC-SR04

Internamente, el sensor está formado por *dos transductores* (que aquí son componentes electrónicos que transforman una magnitud eléctrica en sonido). Cada uno de estos transductores presenta su función: uno es el emisor o Trig y actúa como un *altavoz*, lanzando un pulso de ultrasonidos; y el otro transductor es el receptor o Echo, actuando como un *micrófono*, escuchando el eco que se produce al chocar el pulso de ultrasonido con el objeto y rebotar. Dicho así, podemos pensar que lo que nos muestra no es una distancia sino el tiempo que tarda el pulso en llegar al objeto (chocando) y venir de vuelta para ser captado por el transductor Echo. Pero, si tenemos en cuenta la velocidad a la que se propaga el sonido en el aire (340m/s) y ese tiempo que tarda el eco en llegar al transductor micrófono, aplicado la fórmula del MRU $e=v \times t$ y dividiendo ese espacio entre 2 (ya que estamos contabilizando el de ida y el de vuelta), obtendríamos la distancia a la cual se encuentra el objeto. Esta explicación del funcionamiento del sensor distancia se clarifica con la siguiente imagen:



Por lo tanto, la distancia que nos mide el sensor en metros es:

$$D = \frac{v \times t}{2} = \frac{340 \times t}{2} = 170 \times t \text{ (m)}$$



Un posible programa que simule y controle el sensor distancia para que detecte un objeto puede ser el se crea a continuación. Necesitamos de dos objetos: un objeto arduino que llamo "ArduinoBat" y que es un murciélago y un objeto que se moverá por el escenario atendiendo a la variable "distancia" en el eje X y que llamo "PlofBat".

La siguiente figura muestra ambos objetos en el escenario:



Para su programación se ha tenido en cuenta la programación modificada del firmware que se presenta en el siguiente link de: [TR3SDLAND](https://www.tr3sdland.com/).

Las modificaciones en el firmware que posibilitan el uso de este sensor, le otorgan al pin 10 la patilla Trig y al pin A5 la patilla Echo

El programa que se ha diseñado muestra dos objetos: el murciélago (*ArduinoBat*) y *el objeto que se moverá por el escenario* en el sentido del eje X variando su color según la distancia de aproximación al objeto arduino, de modo que, si se encuentra a su lado, presentará un color rojo de peligro y escucharemos un sonido de forma intermitente. Pero, al estar lejos, el color del objeto se aleja del color rojo (peligro) y no se escucha ningún sonido de peligro por proximidad. En el caso de que este objeto, llamado "*PlofBat*" se esté acercando al objeto arduino (pero no demasiado, un valor de distancia entre 20 y 50) comenzará a avisarnos con un sonido más espaciado en el tiempo y con un color cercano al rojo.

El script para el objeto arduino "ArduinoBat" es el siguiente:



En el pequeño script inferior de la figura anterior se pretende darle sensación de movimiento al objeto arduino a la vez que lo dejamos fijo en la posición lateral izquierda del escenario (-177, 105).

El script para el objeto móvil "PlofBat" es el siguiente:



Programa de nuestro objeto "PlotBar". Susana Oubiña Falcón [\(CC-BY\)](#)

Al pulsar la bandera verde, el objeto se sitúa en el lateral derecho del escenario. A continuación comienza a moverse en el eje X atendiendo al valor de proximidad o distancia del sensor HC-SR04. Para que no se salga del escenario en el eje X se ha ponderado la distancia a un valor máximo y mínimo del eje X mediante la ecuación $(\text{distancia}-60) \times 2$.