

PRÁCTICAS DE ARDUINO

Práctica nº 9: Sonidos con Arduino

¿QUÉ ES UN BUZZER O UN ALTAVOZ?

Un buzzer pasivo o un altavoz son dispositivos que **permiten convertir una señal eléctrica en una onda de sonido**. Estos dispositivos no disponen de electrónica interna, por lo que tenemos que proporcionar una señal eléctrica para conseguir el sonido deseado.

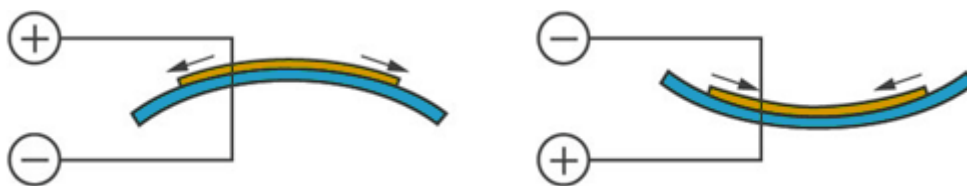
En oposición, los buzzer activos disponen de un oscilador interno, por lo que únicamente tenemos que alimentar el dispositivo para que se produzca el sonido.

Pese a tener la complejidad de proporcionar y controlar nosotros la señal eléctrica, los buzzer pasivos y de los altavoces **tienen la ventaja de que podemos variar el tono emitido** modificando la señal que aplicamos al altavoz, lo que nos permite generar melodías.

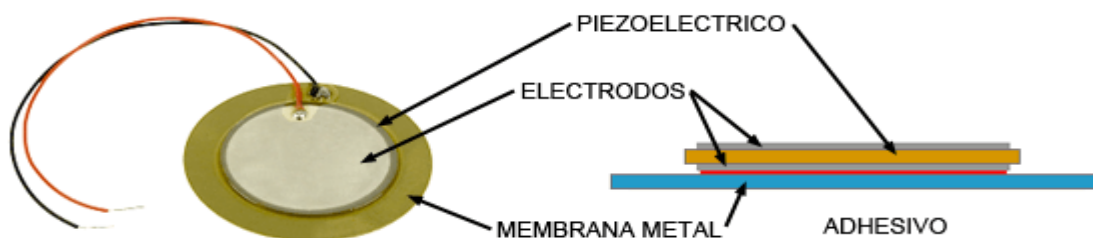
¿CÓMO FUNCIONA UN BUZZER Y UN ALTAVOZ?

Técnicamente tanto buzzers como altavoces son transductores electroacústicos, es decir, **dispositivos que convierten señales eléctricas en sonido**. La diferencia entre ambos es el fenómeno en el que basan su funcionamiento.

Los buzzer son transductores piezoeléctricos. Los materiales piezoeléctricos tienen la propiedad especial de variar su volumen al ser atravesados por corrientes eléctricas.

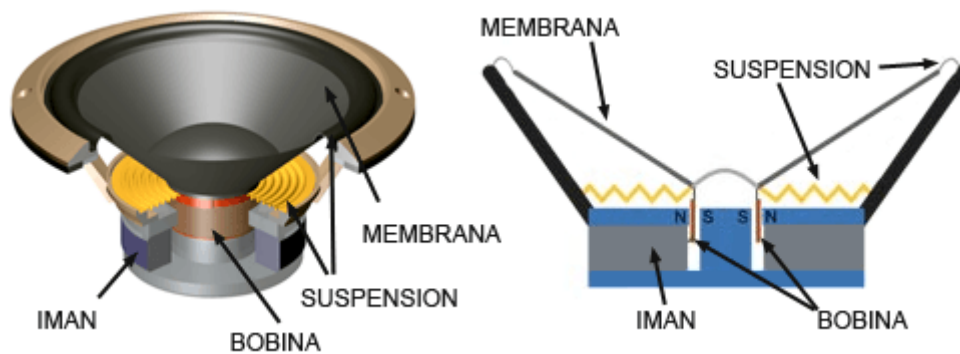


Un buzzer aprovecha este fenómeno para hacer vibrar una membrana al atravesar el material piezoeléctrico con una señal eléctrica.



Los buzzer son dispositivos pequeños y compactos, con alta durabilidad, y bajo consumo eléctrico. Por contra, la calidad de sonido es reducida.

Por su parte, **un altavoz basa su funcionamiento en el magnetismo**. Se dispone de un imán permanente que, normalmente, es fijo a la carcasa. Por otro lado, una bobina móvil se acopla a una membrana flexible.



Al hacer circular una corriente por la bobina el campo magnético resultante genera una atracción con el imán, haciendo vibrar la membrana.

Los altavoces, por presentan una mejor calidad de sonido pero, en general, necesitan mayor potencia y es necesario disponer de dispositivos de amplificación para su uso.

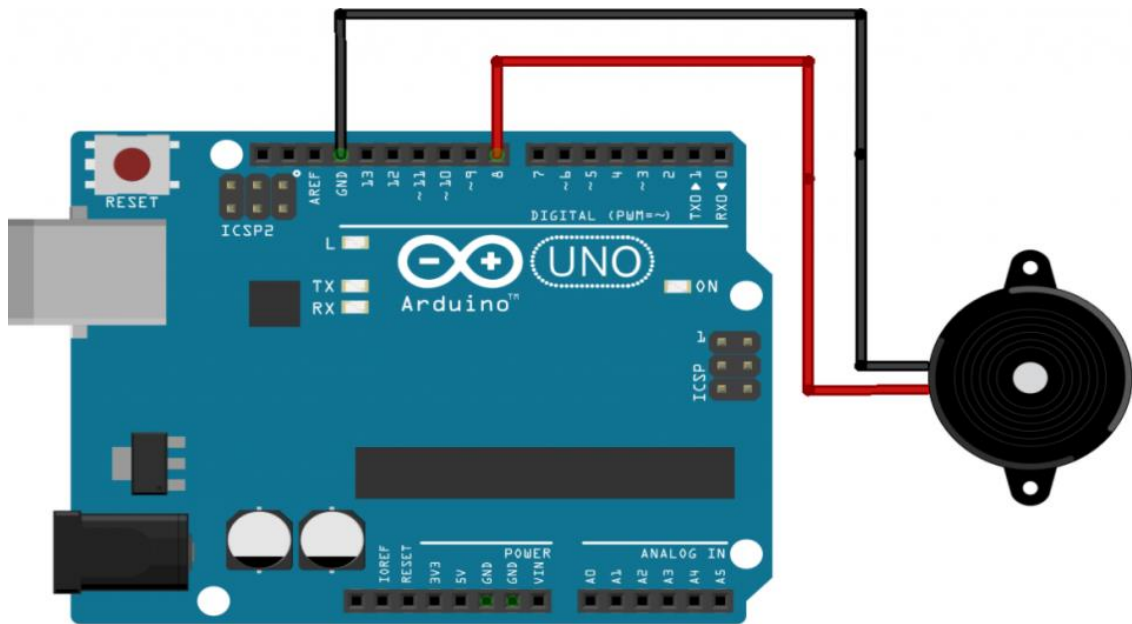
En resumen, un zumbador es un elemento parecido a un altavoz pero sólo emite zumbidos (típico sonido que emiten los electrodomésticos).

MATERIALES

- Arduino UNO.
- Cable USB tipo A-B.
- 1 Buzzer
- Cables de conexión.

MONTAJE

Para hacerlo sonar utilizaremos los pines 8 y GND.



PROGRAMACIÓN

FUNCIÓN TONE

Con Arduino también podemos crear sonidos gracias a la función *tone* cuya sintaxis es la siguiente:

tone(pinsalida,frecuencia);

tone(pinsalida, frecuencia, tiempo)

Esta función lo que hace es intercambiar valores HIGH/LOW a la frecuencia deseada en el pin seleccionado hasta que la volvemos a llamar con otra frecuencia o le ordenamos que pare con la función *noTone*.

FUNCIÓN NOTONE

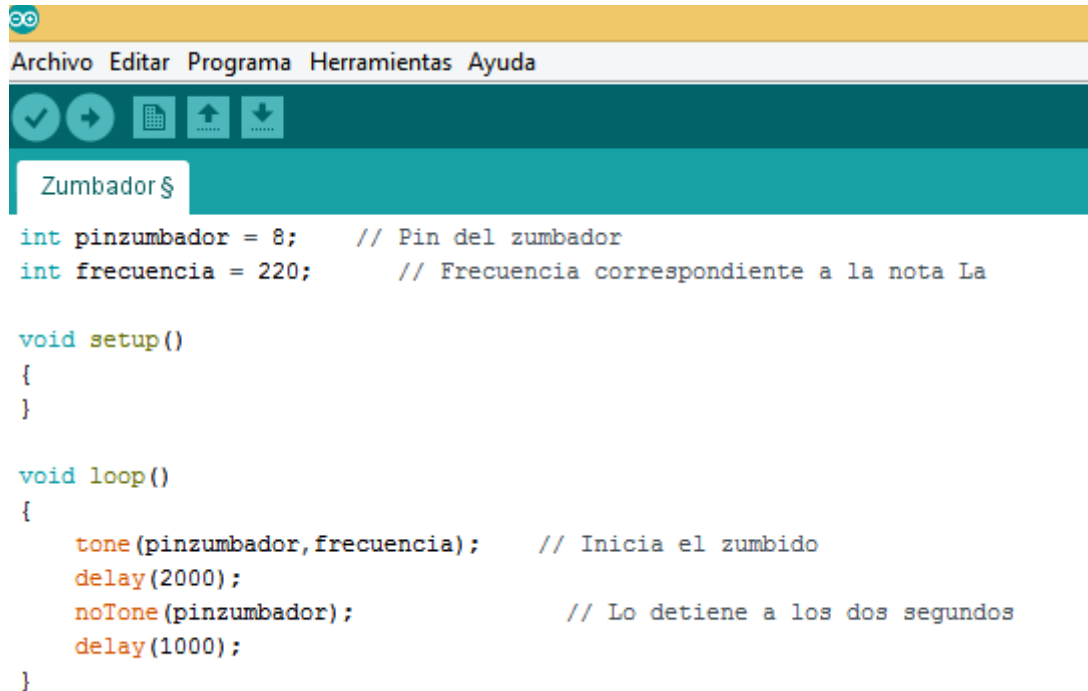
noTone(pinsalida);

Utilizaremos la instrucción *noTone* si no indicamos un tiempo en la instrucción *tone*. Esta instrucción indica que se deje de emitir la señal por el pin indicado.

Tenemos que tener en cuenta que **sólo es capaz de generar un único tono a la vez**. Si llamamos a la función *tone* para que genere sonidos en otro pin sin haber detenido el que está sonando no ocurrirá nada.

EJEMPLO DE CÓDIGO

Una vez realizado el montaje para comprobar su correcto funcionamiento, cargamos el siguiente código:



```
int pinzumbador = 8;    // Pin del zumbador
int frecuencia = 220;    // Frecuencia correspondiente a la nota La

void setup()
{
}

void loop()
{
    tone(pinzumbador,frecuencia);    // Inicia el zumbido
    delay(2000);
    noTone(pinzumbador);              // Lo detiene a los dos segundos
    delay(1000);
}
```

Podemos observar dos cosas:

- Al cargar el programa suena brevemente el zumbador. Esto se debe a que utilizamos el pin 8 y éste se pone a alto (HIGH) brevemente unas veces cuando se inicia la placa.
- Si variamos la frecuencia prácticamente no varía el tono o no suena. La causa de ésto es que un zumbador tiene poca capacidad para reproducir sonidos fielmente. La frecuencia influye tan poco que incluso sonaría conectándolo entre los pines 5V y GND. Para frecuencias demasiado altas el zumbador no responde.

Otros aspectos importantes a destacar:

- Pese a su sencillez, al usar las funciones para la generación de tone tenemos que asumir importantes limitaciones.
- La función Tone emplea el Timer 2, por lo que mientras esté funcionando no podremos usar las salidas PWM en los pines 3 y 11 en Arduino Uno.
- No podemos usar la función tone() en dos pines de forma simultánea. Debemos apagar el tono con la función noTone() antes de poder usarlo en otro pin.
- Los rangos de la función tone son de 31 Hz a 65535 Hz.

Ejercicios propuestos

1. Comprueba que el programa “Zumbador” funciona correctamente en tu placa Arduino.
2. Cambia los valores de la función delay para que el zumbador suene durante más o menos tiempo.
3. Haz que el programa se repita 16 veces.
4. El siguiente código se muestra el uso las funciones vistas en esta práctica es un ejemplo simple, en el que empleamos el mismo montaje de conexión para en el pinBuzzer = 8 generar una función de 440Hz durante un segundo (1000 ms), pararlo durante 500ms, y finalmente un tono de 523Hz durante 300ms, para repetir el programa tras una pausa de 500ms. Modifícalo conectando el buzzer al pin 9, cambiando la frecuencia y duración de cada tono.

```
const int pinBuzzer = 8;

void setup()
{
}

void loop()
{
    //Generar tono de 440Hz durante 1000 ms
    tone(pinBuzzer, 440);
    delay(1000);

    //Detener tono durante 500ms
    noTone(pinBuzzer);
    delay(500);

    //Generar tono de 523Hz durante 500ms, y detenerlo durante 500ms.
    tone(pinBuzzer, 523, 300);
    delay(500);
}
```