

QSI-2610T: BUZZER 12mA/12VCD 3.5KHZ 85dB DRIVER INTERNO



Descripción

Zumbador, buzzer en inglés, es un transductor electroacústico que produce un sonido o zumbido de un mismo tono (generalmente agudo). Sirve como mecanismo de señalización, aviso o alarma y se utiliza en múltiples sistemas, como en automóviles o en electrodomésticos, incluidos los despertadores. Su construcción consta de dos elementos, un electro imán y una lámina metálica de acero. El zumbador es de tipo pasivo y por lo tanto, puede ser conectado a circuitos integrados especiales para así lograr distintos tonos y sonidos intermitentes o continuos ya que permiten convertir una señal eléctrica en una onda de sonido. Cuando se acciona, la corriente pasa por la bobina del electro imán y produce un campo magnético variable que hace vibrar la lámina de acero sobre la armadura.

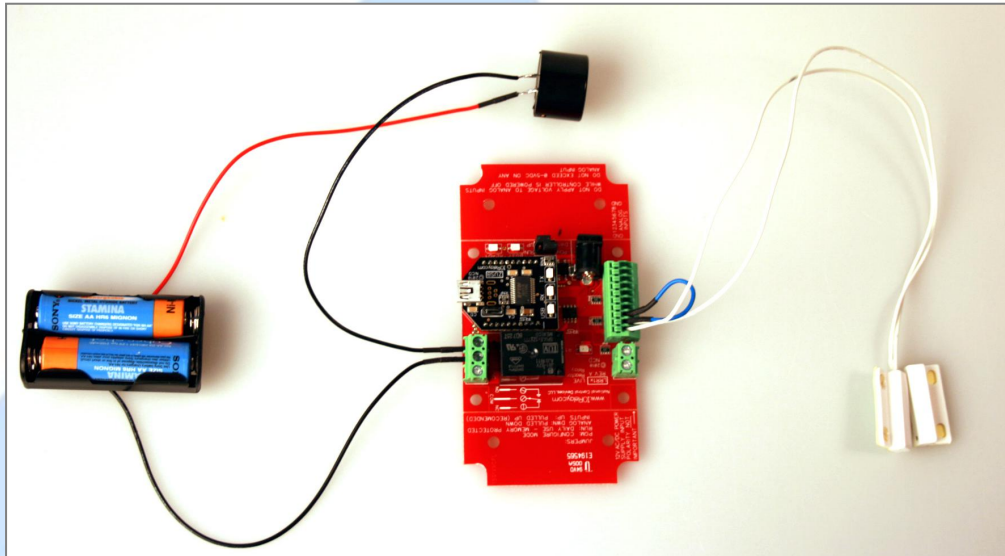
Especificaciones

Tipo de tono: Constante	Alimentación: 12 VCD
Voltaje de operación: 3 ~ 24 VCD	Consumo de corriente: 12 mA
Frecuencia oscilatoria: 3.5 kHz	Nivel de sonido: 85 dB/30cm
Temperatura de funcionamiento: -20 ~ 80°C	Peso: 4 g

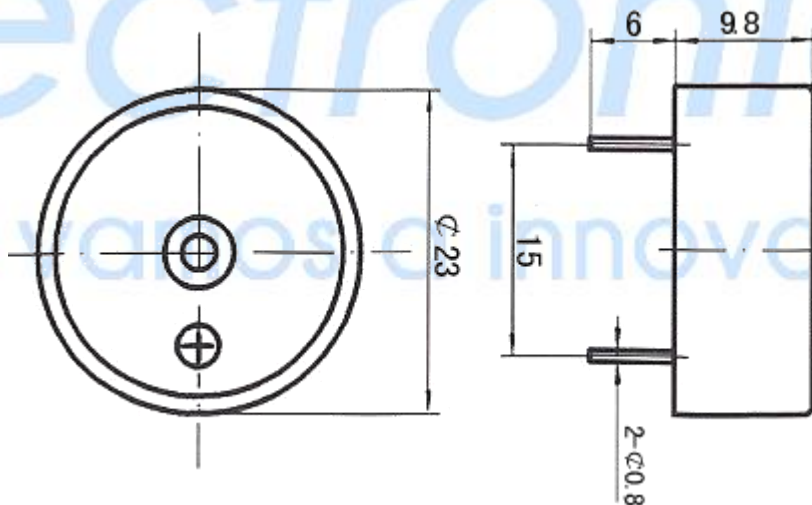
Aplicaciones

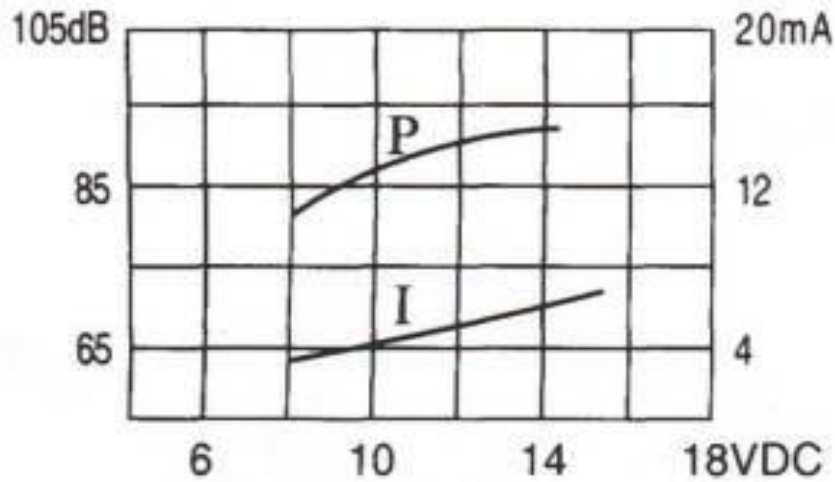
Los generadores de sonido piezoeléctricos y electromagnéticos son dispositivos aptos para el diseño de alarmas y controles acústicos de estrecho rango de frecuencia, por ejemplo, en aparatos domésticos y de medicina.

- Ideal para realizar prototipos.
- proyectos.
- alarmas para puertas o ventanas.
- pruebas, etc.



Dimensiones **Unit:mm** **Tolerance:±0.5**





Ejemplo para poder encender y apagar un LED al ritmo de la música de Star Wars que será reproducida por el zumbador.

Para ello necesitará los siguientes materiales:

- Arduino UNO
- Protoboard
- LED (rojo)
- Resistencia de 300 Ohms
- Buzzer

Código del programa:

```
#define c3 7634 // define es otra manera de declarar constantes en la IDE de
Arduino
#define d3 6803 // todas estas constantes representan un tono de las
diferentes melodías
#define e3 6061
#define f3 5714
#define g3 5102
#define a3 4545
#define b3 4049
#define c4 3816 // 261 Hz
#define d4 3401 // 294 Hz
#define e4 3030 // 329 Hz
#define f4 2865 // 349 Hz
#define g4 2551 // 392 Hz
#define a4 2272 // 440 Hz
#define a4s 2146
#define b4 2028 // 493 Hz
#define c5 1912 // 523 Hz
#define d5 1706
```

```
#define d5s 1608
#define e5 1517 // 659 Hz
#define f5 1433 // 698 Hz
#define g5 1276
#define a5 1136
#define a5s 1073
#define b5 1012
#define c6 955
#define R 0 // nota especial R para representar un silencio o pausa

// Melodía 1: Imperial March tema de Star Wars
int melody1[] = {a4, R, a4, R, a4, R, f4, R, c5, R, a4, R, f4, R, c5, R, a4, R, e5,
R, e5, R, e5, R, f5, R, c5, R, g5, R, f5, R, c5, R, a4, R};
int beats1[] = {50, 20, 50, 20, 50, 20, 40, 5, 20, 5, 60, 10, 40, 5, 20, 5, 60, 80,
50, 20, 50, 20, 50, 20, 40, 5, 20, 5, 60, 10, 40, 5, 20, 5, 60, 40};

// Melodía 2: Tema oficial de Star Wars
int melody2[] = {f4, f4, f4, a4s, f5, d5s, d5, c5, a5s, f5, d5s, d5, c5, a5s, f5,
d5s, d5, d5s, c5};
int beats2[] = {21, 21, 21, 128, 128, 21, 21, 21, 128, 64, 21, 21, 21, 128, 64, 21,
21, 21, 128 };

int MAX_COUNT = sizeof(melody1) / 2;
long tempo = 10000;
int pause = 1000;
int rest_count = 50;
int toneM = 0;
int beat = 0;
long duration = 0;
int potVal = 0;
int speakerOut = 9;
int led = 6;
//declaración de variables

void setup() {
    pinMode(speakerOut, OUTPUT);
    pinMode(led, OUTPUT);
    Serial.begin(9600);
}
```

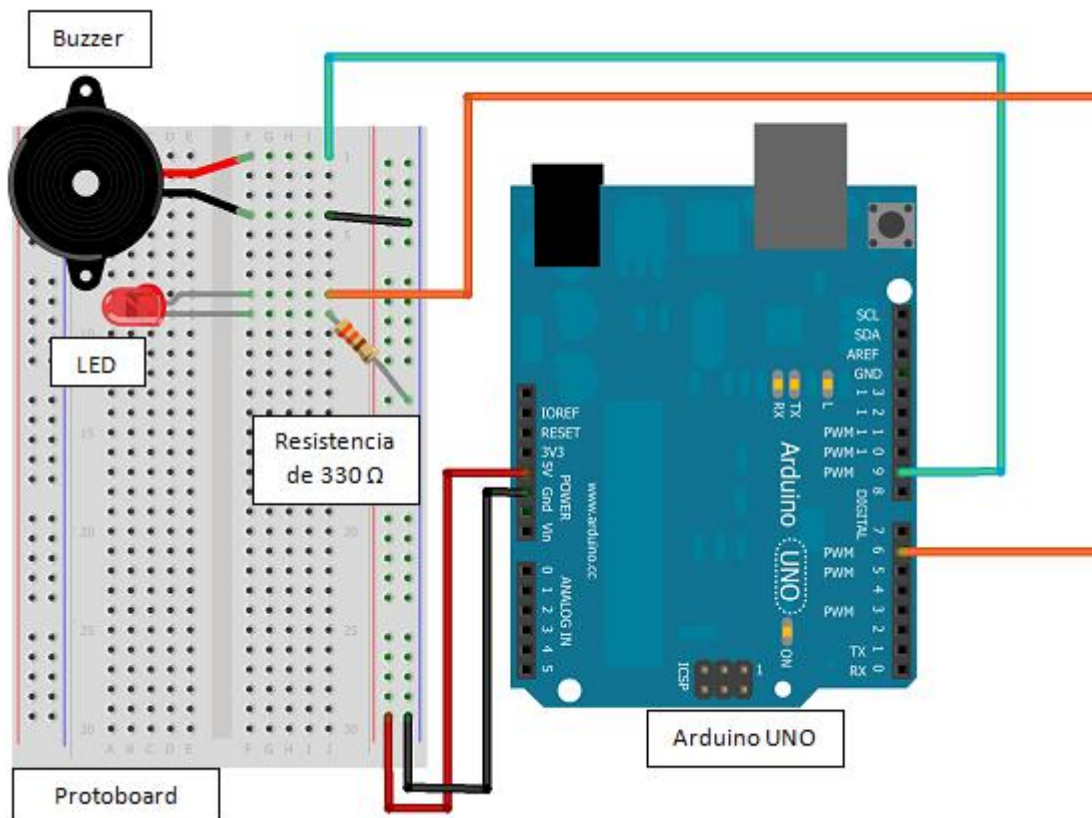
```
void loop() {
  // Melodía 1
  for (int i=0; i<MAX_COUNT; i++) {
    toneM = melody1[i];
    beat = beats1[i];
    duration = beat * tempo;
    playTone();
    delayMicroseconds(pause);
  }
  // Melodía 2
  MAX_COUNT = sizeof(melody2)/2;

  for (int i = 0; i<MAX_COUNT; i++) {
    toneM = melody2[i];
    beat = beats2[i];
    duration = beat*tempo;
    playTone();
    delayMicroseconds(pause);
  }
}

void playTone() {
  long elapsed_time = 0;
  if (toneM > 0) {
    digitalWrite(led,HIGH);
    while (elapsed_time < duration) {
      digitalWrite(speakerOut,HIGH);
      delayMicroseconds(toneM / 2);
      digitalWrite(speakerOut, LOW);
      delayMicroseconds(toneM / 2);
      elapsed_time += (toneM);
    }
    digitalWrite(led,LOW);
  }else{
    for (int j = 0; j < rest_count; j++) {
      delayMicroseconds(duration);
    }
  }
}
```

Diagrama de conexión

En la siguiente imagen observamos la conexión, por la cual cargamos el programa (o sketch) al Arduino para que haga zumbar su buzzer con el tono de Star Wars. Tenga en cuenta que las conexiones pueden ser intercambiadas dentro de los parámetros de las funciones.



Electrónica

¿Qué vamos a innovar hoy?

	AG Electrónica S.A.P.I. de C.V. República del Salvador N° 20 Segundo Piso Teléfono: (01)55 5130 - 7210		
ACOTACIÓN: N/A	http://www.agelectronica.com	ESCALA: N/A	REALIZO: DCR REV:DMSP
TOLERANCIA: N/A	BUZZER 12mA/12VDC 3.5KHZ 85dB DRIVER INTERNO		
TOLERANCIA: N/A	Fecha: 05/03/2019	No. Parte: QSI-2610T	