

# BUZZER ACTIVO DE 9V

OKY0151-9V



Productos evaluados por **ingenieros calificados**



**Garantía y seguridad** en cada producto



Experiencia de compra en la **calidad** como sello distintivo

## Descripción

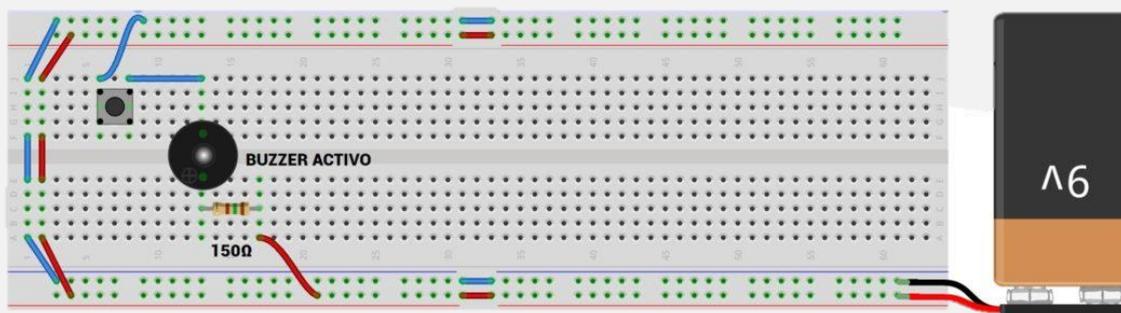
El OKY0151 es un buzzer activo de 9V, que permite convertir la señal eléctrica en acústica, normalmente utilizado como alarma o rastreador.

### Especificaciones técnicas

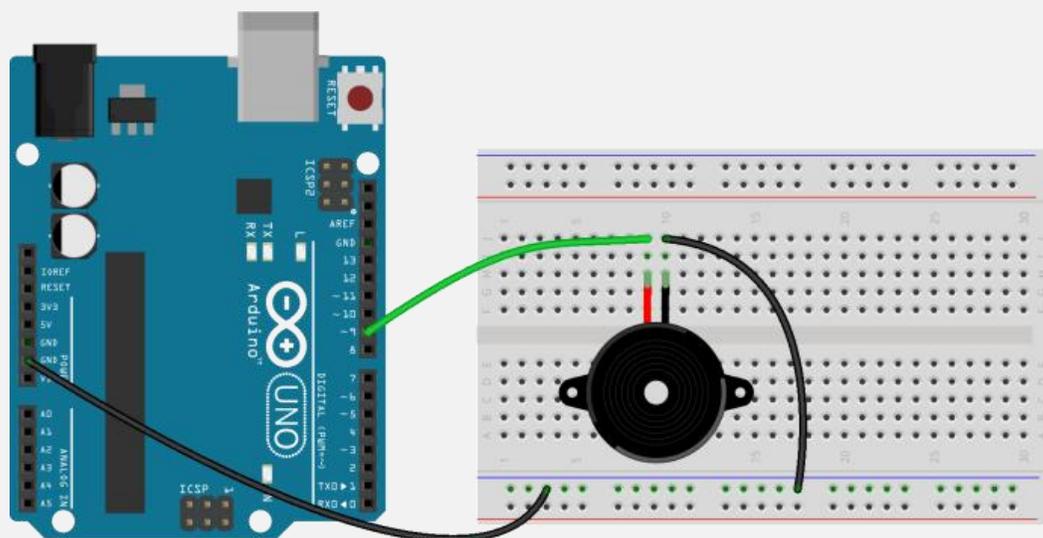
VCC	9 VDC
Corriente	25mA
Frecuencia	2300 Hz $\pm$ 200Hz
Dimensiones	12 mm x 9.5 mm x 7.5 mm
Salida mínima de sonido a 10cm	90 dB
Temperatura de operación	-30°C a 85°C

## Ejemplo de conexión en protoboard

En el siguiente ejemplo de conexión se utiliza el buzzer con un botón y una batería de 9V, se utiliza una resistencia para limitar la corriente proporcionada, y en este caso el botón es quien permitirá el paso de la corriente que a su vez activará el buzzer traduciendo a sonido.



## Ejemplo de conexión y aplicación con Arduino



Componentes necesarios:

- Arduino (Uno, Nano, Mega, etc.)
- Buzzer activo de 9V
- Cables de conexión
- Protoboard (opcional)

Conexiones:

- Conecta el terminal positivo del buzzer al pin digital (por ejemplo, el pin 9) del Arduino.
- Conecta el terminal negativo del buzzer GND.

Código:

Este código mandará la señal al buzzer para generar un tono, y a su vez hace que cada vez que el ciclo se repita, la frecuencia del buzzer cambie, generando diferentes tono en un rango de frecuencia de 100 a 2000

```
// Definimos el pin al que está conectado el buzzer
const int buzzerPin = 9;

// Variables para la frecuencia
int minFrequency = 100; // Frecuencia mínima en Hz
int maxFrequency = 2000; // Frecuencia máxima en Hz
int stepFrequency = 50; // Paso de frecuencia

void setup() {
  // Configuramos el pin del buzzer como salida
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  // Ciclo para aumentar la frecuencia
  for (int freq = minFrequency; freq <= maxFrequency; freq += stepFrequency) {
    tone(buzzerPin, freq); // Genera el tono en la frecuencia actual
    delay(100); // Espera 100 milisegundos
  }

  // Ciclo para disminuir la frecuencia
  for (int freq = maxFrequency; freq >= minFrequency; freq -= stepFrequency) {
    tone(buzzerPin, freq); // Genera el tono en la frecuencia actual
    delay(100); // Espera 100 milisegundos
  }
}
```

## Explicación del Código:

- Definición de Pines y Variables:

- buzzerPin define el pin al que está conectado el buzzer.
- minFrequency y maxFrequency definen los límites de la frecuencia.
- stepFrequency define el incremento o decremento de la frecuencia en cada paso del ciclo.

- Configuración Inicial

-pinMode(buzzerPin, OUTPUT); configura el pin del buzzer como salida.

- Ciclo Principal:

- Dos ciclos for se utilizan para variar la frecuencia del buzzer.
- El primer ciclo aumenta la frecuencia desde minFrequency hasta maxFrequency.
- El segundo ciclo disminuye la frecuencia desde maxFrequency hasta minFrequency.

## Enlace externo: Buzzer activo vs Buzzer pasivo y aplicaciones con Arduino

Kriss Electronics. (2022, 24 agosto). *Buzzer Activo vs. Buzzer Pasivo | CREA Melodías con ARDUINO!* [Video]. YouTube. <https://www.youtube.com/watch?v=x5JO6w-kl10>

**AG Electrónica SAPI de CV**  
República de El Salvador 20 Piso 2, Centro  
Histórico, Centro, 06000 Ciudad de México,  
CDMX  
Teléfono: 55 5130 7210

Realizó Adrián Jesús Beltrán Cruz

Revisó Ing. Jesús Daniel Ibarra Noguez

Fecha 11/07/2024

