

**Parte 2**

# Lección

# 5

**Activar zumbador**

## Resumen

En esta lección, aprenderá a generar un sonido con un timbre activo.

### Componente necesario:

- (1) x Elegoo Uno R3
- (1) x zumbador de Active
- (2) x F M cables (cables de hembra a macho DuPont)

## Introducción del componente

### ZUMBADOR:

**Zumbadores** electrónicos son alimentados por CC, equipado con un circuito integrado. Son ampliamente utilizados en computadoras, impresoras, fotocopadoras, alarmas, juguetes electrónicos, dispositivos electrónicos automotores, teléfonos, alarmas y otros productos electrónicos para dispositivos de voz. ZUMBADORES se pueden categorizar como activa y pasiva. Gire que los pernos de dos zumbadores boca arriba. Con un tablero de circuito verde es un zumbador pasivo, mientras que el otro cerrado con una cinta negra es un activo.

**La** diferencia entre los dos es que un buzzer activo tiene una fuente oscilante integrada, por lo que va a generar un sonido cuando se electrifica. Una alarma pasiva no tiene esa fuente para que no pitan si se utilizan señales de DC; en cambio, necesitará usar ondas cuadradas cuya frecuencia es entre 2K y 5K manejarlo. El zumbador activo es a menudo más caros que el pasivo debido a múltiples circuitos oscilantes incorporados.



# Conexión Esquema

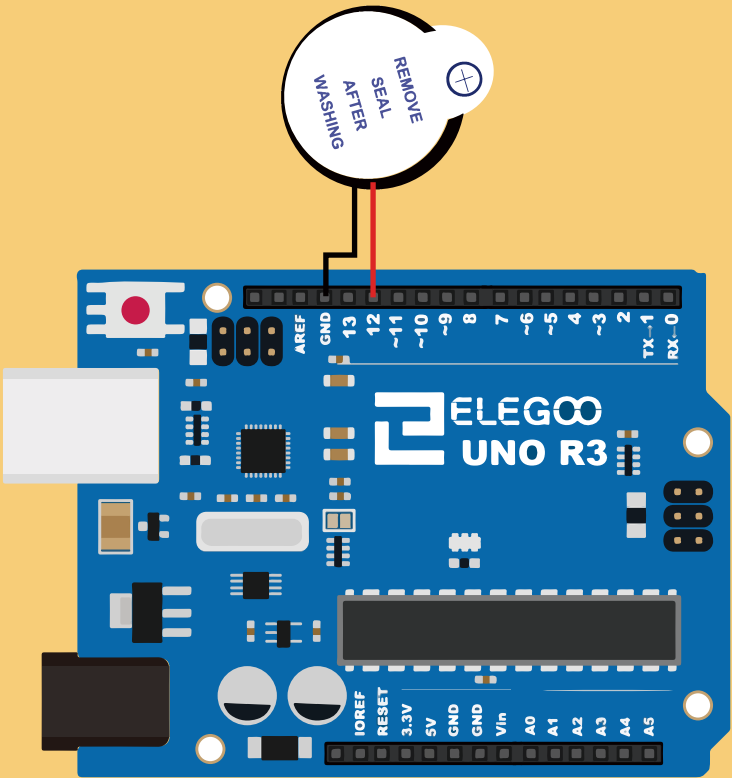
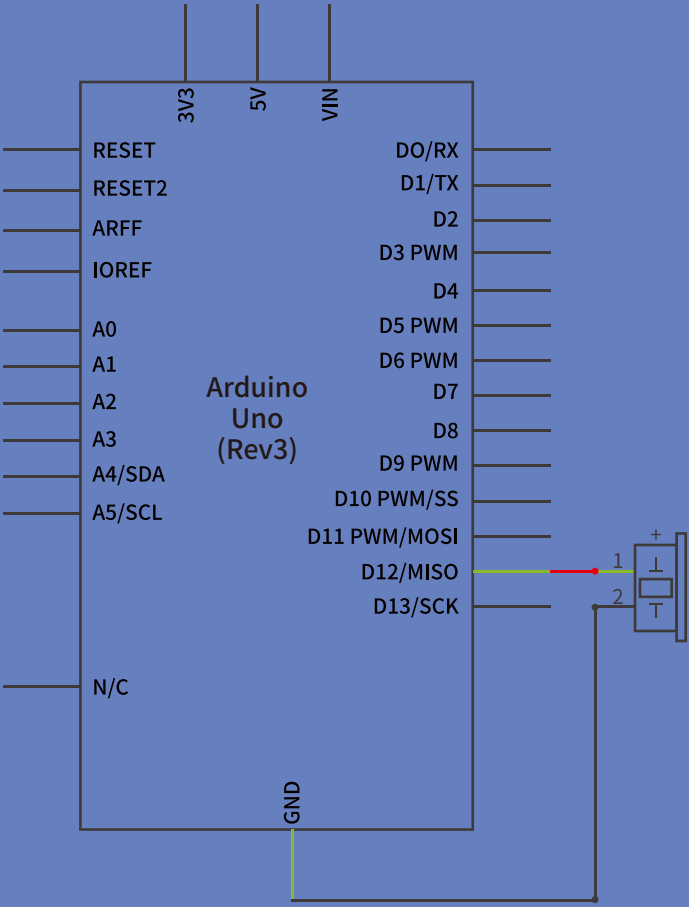



Diagrama de conexiones

# Código

Abra el programa.

c:\computer > software (D:) > project > Course > Elegoo The Most Complete Starter Kit for UNO > English > Part 2 Module learning > 2.5 Active buzzer > active				
Name	Date modified	Type	Size	
 active	12/8/2016 9:32 PM	INO File	1 KB	

## while(condición)

### [Estructura de control]

#### Descripción

Un bucle “While” se repetirá continua e infinitamente, hasta que la expresión dentro del paréntesis, () se vuelva falsa. Algo debe modificar la variable comprobada, o el ciclo “While” nunca terminará. Esto podría ser en su código, una variable incrementada, o una condición externa, como el resultado de un sensor.

```
while(1)
{
    .....
}
```

#### Parámetros

**condición:** una expresión booleana que se evalúa como verdadera o falsa.

#### Sintaxis

```
while (condición) {
    // declaracion(es)
}
```

```
unsigned char i;
```

#### unsigned char

##### [Tipos de datos]

Range: 0~255

#### Descripción

Un tipo de datos sin asignación que ocupa 1 byte de memoria. Igual que el tipo de datos “Byte”.

El tipo de datos sin “unsigned char” soporta números del 0 al 255.

Para mantener la coherencia del estilo de programación de Arduino, se debe escoger el tipo de .

#### Sintaxis

```
unsigned char var = val;
```

#### Parámetros

**var:** nombre de la variable.

**val:** el valor asignado a esa variable.

#### Código de ejemplo

```
unsigned char myChar = 240;
```