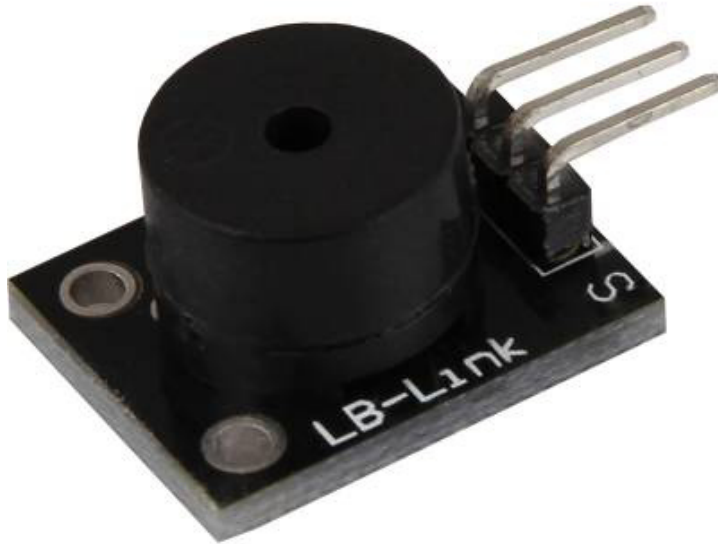


KY-006 Passiv Piezo-Buzzer module

Contents

1 Picture	1
2 Technical data / Short description	1
3 Pinout	2
4 Code example Arduino	2
5 Code example Raspberry Pi	3

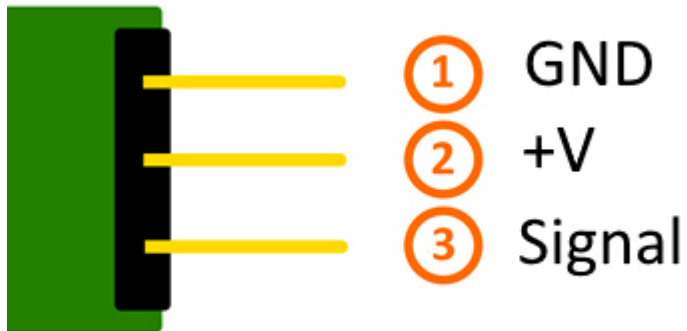
Picture



Technical data / Short description

PWM-Signals of different frequencies can be used to create different sounds from the Piezo-Buzzer.

Pinout



Code example Arduino

This is an example program which will start an alarm signal on the buzzer via square wave voltage.

```
int buzzer = 8 ; // Declaration of the buzzer-output pin

void setup ()
{
  pinMode (buzzer, OUTPUT) ;// Initialization of the output pin.
}

void loop ()
{
  unsigned char i;
  while (1)
  {
    // The buzzer will be controlled by 2 different frequencies in this program.
    // The signal is a square wave signal.
    // The on and off of the buzzer will generate a sound which is nearly the sound of the frequency.
    // The frequency will be defined from the time of the on and off period.

    //Frequency 1
    for (i = 0; i <80; i++)
    {
      digitalWrite (buzzer, HIGH) ;
      delay (1) ;
      digitalWrite (buzzer, LOW) ;
      delay (1) ;
    }
    //Frequency 2
    for (i = 0; i <100; i++)
    {
      digitalWrite (buzzer, HIGH) ;
      delay (2) ;
      digitalWrite (buzzer, LOW) ;
      delay (2) ;
    }
  }
}
```

KY-006 Passiv Piezo-Buzzer module

Connections Arduino:

Sensor signal = [Pin 8]
Sensor - = [Pin GND]

Example program download[KY-006_Buzzer](#)**Code example Raspberry Pi**

This example program uses software-PWM, to generate a square wave with defined frequency.

The buzzer will generate a sound which is nearly the sound of the square wave frequency.

```
#Needed modules will be imported and configured.
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

#The output pin, which is connected with the buzzer, will be declared here.
GPIO_PIN = 24
GPIO.setup(GPIO_PIN, GPIO.OUT)
#The software-PWM module will be initialized - a frequency of 500Hz will be taken as default.
Frequenz = 500 #In Hertz
pwm = GPIO.PWM(GPIO_PIN, Frequenz)
pwm.start(50)
# The program will wait for the input of a new PWM-frequency from the user.
# Until then, the buzzer will be used with the before inputted frequency (default 500Hz).
try:
    while(True):
        print "-----"
        print "Current frequency: %d" % Frequenz
        Frequenz = input("Please input a new frequency (50-5000):")
        pwm.ChangeFrequency(Frequenz)

# Scavenging work after the end of the program.
except KeyboardInterrupt:
    GPIO.cleanup()
```

Connections Raspberry Pi:

Signal = GPIO24 [Pin 18]
+V = 3,3V [Pin 1]
GND = GND [Pin 6]

Example program download[KY-006-RPI_PWM](#)

To start, enter the command:

```
sudo python KY-006-RPI_PWM.py
```