

KY-006 Passiv Piezo-Buzzer module

Revision as of 16:27, 20 February 2017 (view source)

[Webmaster Wiki](#) ([Talk](#) | [contribs](#))

(Created page with "===Picture=== none ===Technical data / Short description=== PWM-Signals of different frequencies can be used to get different sounds from the Piezo-Buzzer. =...")

[Newer edit](#) →

(No difference)

Revision as of 16:27, 20 February 2017

Contents

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

Picture

none

Technical data / Short description

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

Pinout

none

Code example Arduino

This is an example program which let 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)
  {
```

KY-006 Passiv Piezo-Buzzer module

```
// 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
// 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) ;
}
}
```

Connections Arduino:

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

Example program download

[KY-006_Buzzer.zip](#)

Code example Raspberry Pi

Program example in the language python

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

The on and off of 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)
<br />#The software-PWM module will be initialized - a frequency of 500Hz will be taken as
Frequenz = 500 #In Hertz
pwm = GPIO.PWM(GPIO_PIN, Frequenz)
pwm.start(50)
<br /># The program will wait for the input of a new PWM-frequency from the user.<br /># U
try:
    while(True):
        print "-----"
        print "Current frequency: %d" % Frequenz
        Frequenz = input("Please input a new frequency (50-5000):")
```

KY-006 Passiv Piezo-Buzzer module

```
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.zip](#)

To start with the command:

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