

KY-006 Passives Piezo-Buzzer Modul

Inhaltsverzeichnis

1 Bild	1
2 Technische Daten / Kurzbeschreibung	1
3 Pin-Belegung	2
4 Codebeispiel Arduino	2
5 Codebeispiel Raspberry Pi	3

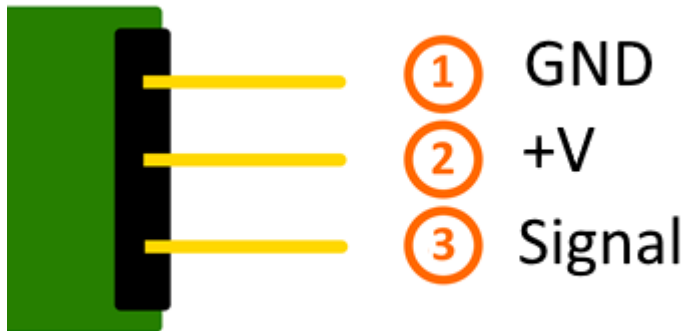
Bild



Technische Daten / Kurzbeschreibung

Mit PWM-Signalen verschiedener Frequenzen angesteuert, können mit dem passiven Piezo-Buzzer verschiedene Töne erzeugt werden.

Pin-Belegung



Codebeispiel Arduino

Hier bei handelt es sich um ein Beispielprogramm, welches am Buzzer ein Alarmsignal mittels einer Rechteckspannung erzeugt.

```
int buzzer = 8 ; // Deklaration des Buzzer-Ausgangspin

void setup ()
{
  pinMode (buzzer, OUTPUT) ;// Initialisierung als Ausgangspin
}

void loop ()
{
  unsigned char i;
  while (1)
  {
    // In diesem Programm wird der Buzzer abwechselnd mit zwei verschiedenen Frequenzen an
    // Das Signal hierbei besteht aus einer Rechteckspannung.
    // Das an- und ausmachen am Buzzer generiert dann einen Ton, der in etwa der Frequenz
    // Die Frequenz definiert sich dadurch, wie lang jeweils die An- und Ausphase sind

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

KY-006 Passives Piezo-Buzzer Modul

```

        delay (2) ;
        digitalWrite (buzzer, LOW) ;
        delay (2) ;
    }
}
}

```

Anschlussbelegung Arduino:

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

Beispielprogramm Download

[KY-006_Buzzer.zip](#)

Codebeispiel Raspberry Pi

Programmierbeispiel in der Programmiersprache Python

Das Beispielprogramm nutzt Software-PWM, um am Ausgangspin eine Rechteckspannung mit definierbarer Frequenz zu erstellen.

Durch das An- und Ausschalten wird am Buzzer ein Ton erzeugt, der in etwa der Frequenz der Rechteckspannung entspricht.

```

# Benoetigte Module werden importiert und eingerichtet
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

# Hier wird der Ausgangs-Pin deklariert, an dem der Buzzer angeschlossen ist.
GPIO_PIN = 24
GPIO.setup(GPIO_PIN, GPIO.OUT)

# Das Software-PWM Modul wird initialisiert - hierbei wird die Frequenz 500Hz als Startwert
Frequenz = 500 #In Hertz
pwm = GPIO.PWM(GPIO_PIN, Frequenz)
pwm.start(50)

# Das Programm wartet auf die Eingabe einer neuen PWM-Frequenz vom Benutzer.
# Bis dahin wird der Buzzer mit der vorher eingegebenen Frequenz betrieben (Startwert 500Hz)
try:
    while(True):
        print "-----"
        print "Aktuelle Frequenz: %d" % Frequenz
        Frequenz = input("Bitte neue Frequenz eingeben (50-5000):")
        pwm.ChangeFrequency(Frequenz)

# Aufräumarbeiten nachdem das Programm beendet wurde
except KeyboardInterrupt:
    GPIO.cleanup()

```

Anschlussbelegung Raspberry Pi:

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

KY-006 Passives Piezo-Buzzer Modul

GND = Masse [Pin 6]

Beispielprogramm Download

[KY-006-RPI_PWM.zip](#)

Zu starten mit dem Befehl:

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