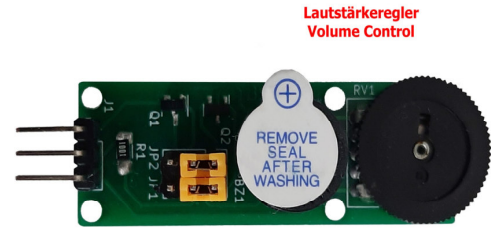




Datenblatt für: net4web AABHL2 Für Arduino Aktiver Buzzer 5V- High/Low umschaltbar mit Abschaltbarer Lautstärkeregelung

Artikelnummer: 0711906114387



High/Low-Schaltbar

Stand: 26.06.2023

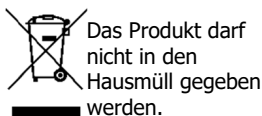
Der net4web AABHL2 wird komplett in Deutschland entwickelt und produziert.

Inhaltsverzeichnis

Beschreibung:	2
Produktfotos:	3
Schaltplan:	4
Platinenmaße:	4
Technische Daten des Aktiven Buzzers:	5
Sketch:	6

Besuchen Sie uns im Internet: www.net4web.de

Im Bereich **Downloads** finden Sie die Konformitätserklärungen zu RoHS und CE, sowie möglicherweise weitere technische Informationen zu diesem Produkt.



Das Produkt erfüllt die Forderungen der EU bzgl. der RoHS-Richtlinie.



Das Produkt erfüllt die Forderungen der EU bzgl. elektromagnetischer Verträglichkeit.

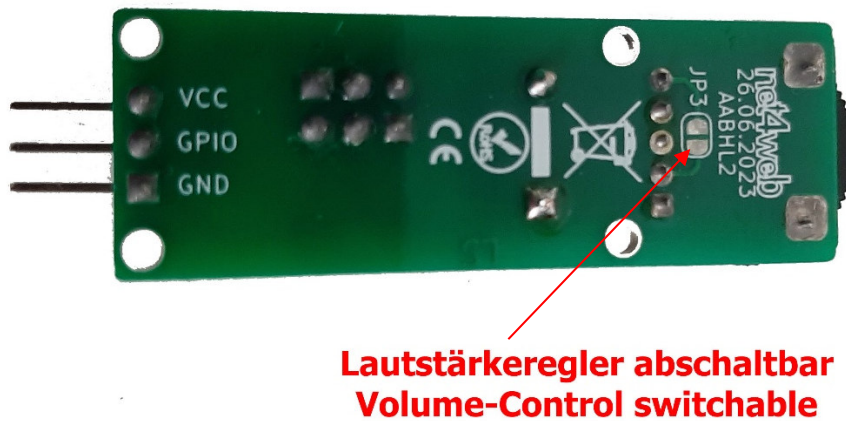
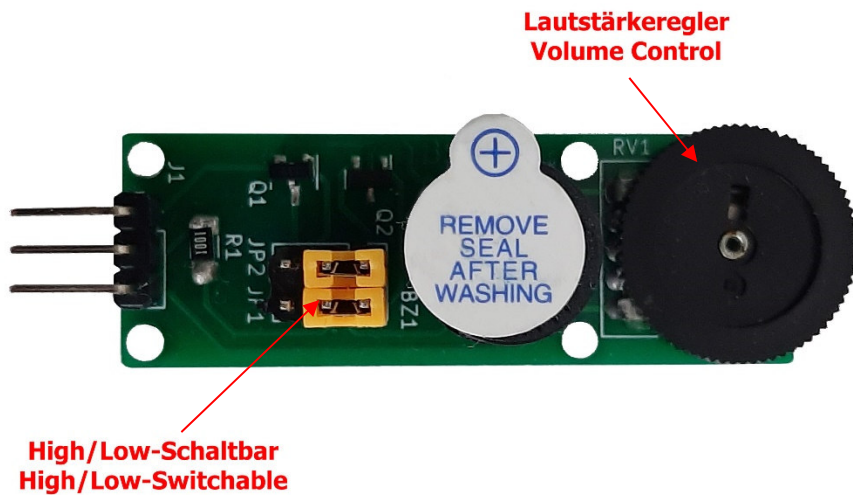
Beschreibung:

Der net4web AABHL2 dient dazu über einen Arduino oder Raspberry Pi einen aktiven Buzzer anzusteuern. Bei vielen Arduino-Nano Clones könnten Sie den Buzzer eventuell direkt an einen digitalen Ausgangspin anschließen. Die Ausgangsleistung reicht dort oft aus (Sollten Sie natürlich vorher prüfen!). Bei anderen Boards, wie etwa ESP32 Clones, sollten Sie unbedingt einen Transistortreiber vorschalten, um Fehlfunktionen zu vermeiden. Die hier vorgestellte Schaltung verfügt bereits über die benötigten Transistortreiber. Aber damit nicht genug! Sie können über die Jumper JP1 und JP2 einstellen, ob Sie den Buzzer mit einem High-Signal oder einem Low-Signal ansteuern. Weil einem in der Entwicklungsphase das ständige Piepsen des Buzzers ganz schön auf die Nerven gehen kann, haben wir eine Lautstärkeregelung vorgesehen. Durch den relativ breiten Betriebsspannungsbereich von ca. +4 bis +7V kann der Buzzer, über das Flachpotentiometer RV1, prima an die gewünschte Lautstärke angepasst werden. Wenn Sie die Lautstärkeregelung nicht oder nicht mehr benötigen, können Sie auf der Lötseite die Lötbrücke JP3 schließen. Damit wird das Potentiometer überbrückt.

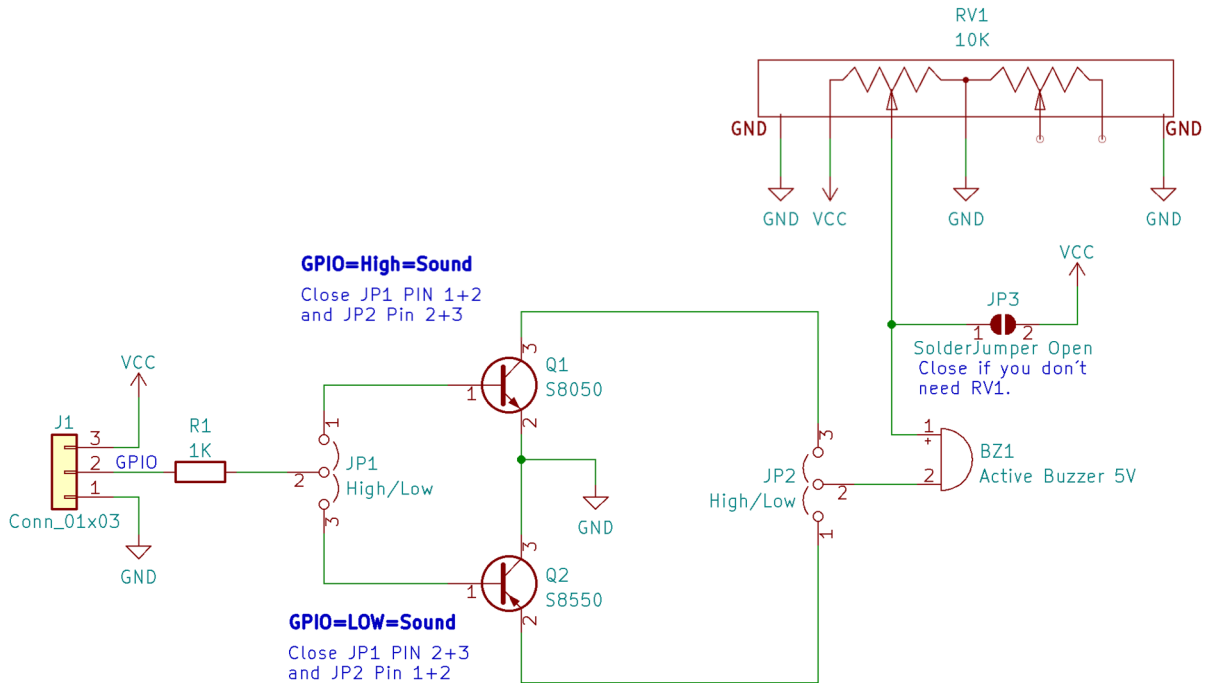
Aktive Buzzer dienen der Signalgebung. Liegt eine Spannung an, bei dem hier angebotenen Modell +5V, geben sie einen gleichmäßigen, durchdringenden, lauten Ton von sich. Und zwar so lange, wie die Spannung anliegt und Strom fließt. Durch den relativ breiten Betriebsspannungsbereich von ca. 4-7V arbeitet der Buzzer auch dann noch, wenn sich z.B. die Batteriespannung absenkt. Aktive Buzzer sind leicht magnetisch. Das sollte man beim Schaltungsdesign beachten, sofern magnetisch beeinflussbare weitere Komponenten, wie etwa ein Hallsensor, verwendet werden.

Neben diesem ausführlichen Datenblatt können Sie auf unserer Homepage auch einen kleinen, einfachen und gut dokumentierten Sketch für Arduino herunterladen, mit dem wir selbst unsere Buzzer-Anwendungen testen. Sie können aber auch gleich den Code am Ende dieses Datenblattes in Ihre Anwendung kopieren. Wenn Sie sich den Sketch ansehen, werden Sie feststellen, dass wir die Zeiten für High und Low Ansteuerung deutlich unterschiedlich lang gemacht haben. Beim Testen hat das den Vorteil, dass man das Gehirn nicht einschalten muss, wenn man den net4web AABHL2 von Low- auf High-Betrieb umjumpert. Man muss eigentlich nur hinhören 😊.

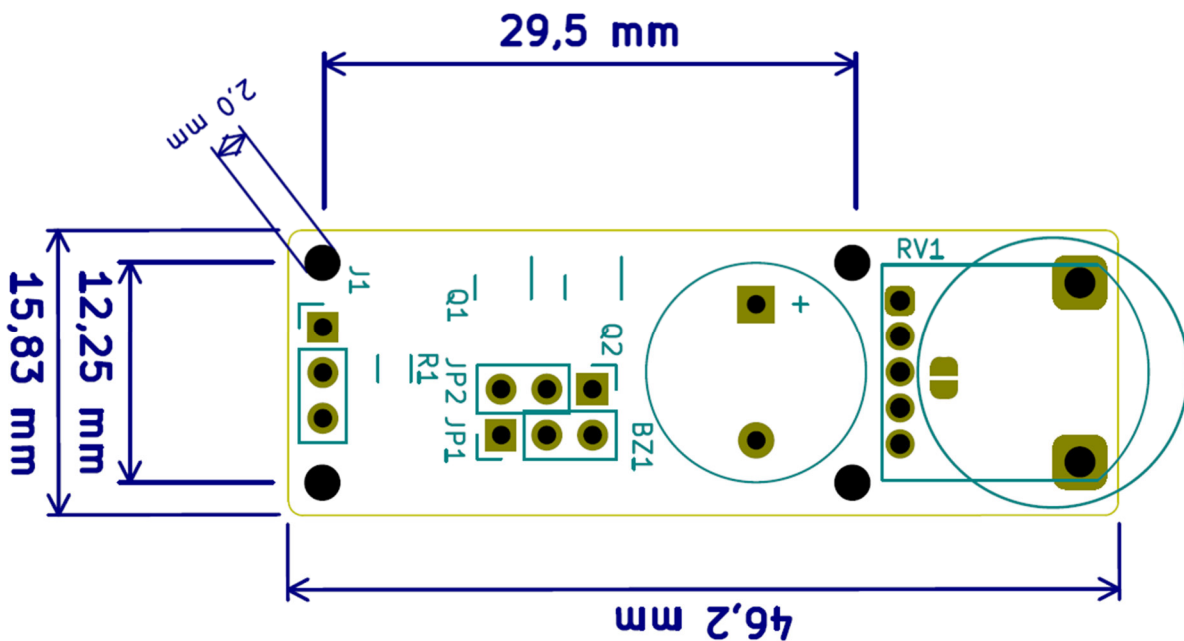
Produktfotos:



Schaltplan:



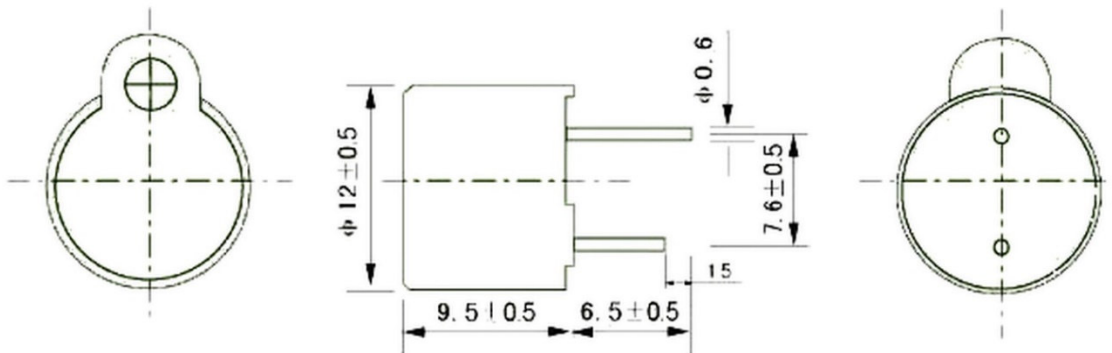
Platinenmaße:



Technische Daten des Aktiven Buzzers:

Durchmesser	12mm
Höhe	9,5mm
Gewicht	2g
Farbe	Schwarz
Optimale Spannung	+5V
Betriebsspannungsbereich	Ca. +3V - +7V
Stromaufnahme	Max. 30mA
Laustärke	>85dB bei +7V, ~79dB bei +5V
Resonanzfrequenz	2300Hz ±500Hz
Rastermaß	7,6mm
Pindurchmesser	0,6mm
Betriebstemperatur	-20°C bis +60°C
Lagertemperatur	-30°C bis +85°C

Alle angegebenen Maße sind ca-Werte und können geringfügig abweichen.





Sketch:

[code]

/******

File name: activebuzzer.ino

Information: Set D4 HIGH/LOW for testing an Active Buzzer.

We have set the delay times to two different values for testing the high/low switch on our AABHL2-PCBs.

So we don't have to think when testing, just listen :-)

Website: <https://www.net4web.de>

E-mail: info@net4web.de

Author: Franz Hansmann

Date: 2023/05/17

*****/

```
int buzzerGPIO=4; //using pin D4 as digital output for the buzzer
```

```
void setup()
```

```
{
```

```
  pinMode(buzzerGPIO,OUTPUT); //Set pin D4 port mode to output
```

```
}
```

```
void loop()
```

```
{
```

```
  digitalWrite(buzzerGPIO,HIGH); //pin D4 set HIGH
```

```
  delay(3000); //delay time, 3000ms
```

```
  digitalWrite(buzzerGPIO,LOW); //pin D4 set LOW
```

```
  delay(1000); //delay time, 1000ms
```

```
}
```

[/code]