



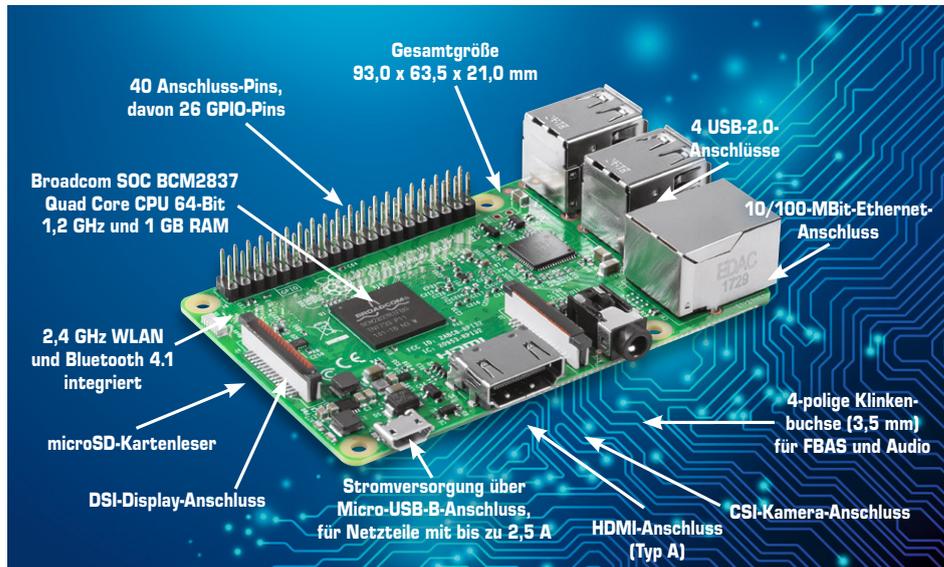
Raspberry Pi

Teil 10: Akustiksensoren – Tanzbär mit dem Raspberry Pi

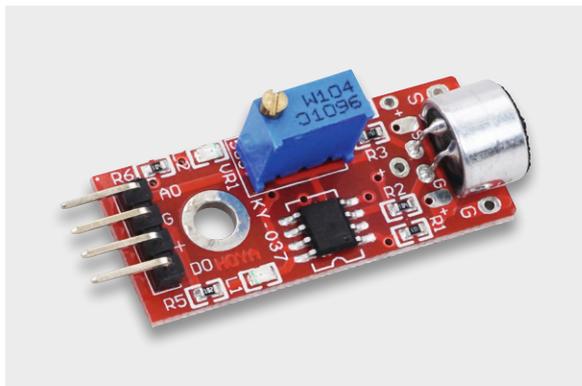
Artikelserie

im ELV Shop

#10036



Mit einem Akustiksensoren lässt sich in der Praxis allerhand anstellen. Angefangen von einer Alarmanlage wie einem Glasbruchmelder über einen Kamera-Auslöser für Speed-Aufnahmen bis hin zum Klingelsensor an der Haustür haben Sie zig Möglichkeiten, diesen günstigen Sensor mit dem Raspberry Pi zu nutzen. In Verbindung mit einem Robotikgestell lässt sich ein Akustiksensoren z. B. für die Steuerung der Motoren verwenden; kommt es beispielsweise zu einem Aufprallgeräusch, dann werden die Motoren automatisch gestoppt. Oder umgekehrt: Soll der Roboter Tanzbär spielen und sich bei vorhandenen Geräuschen bewegen, dann ließe sich so etwas in wenigen Minuten umsetzen.



Passend zu den in der Raspberry-Pi-Artikelserie vorkommenden Sensormodulen bietet ELV ein umfangreiches Sensor-Set an, in dem 40 unterschiedliche Sensoren enthalten sind: www.elv.de: Webcode #10233

Klein, leicht und kompakt: Der Akustiksensoren lässt sich gerade für mobile Zwecke gut einsetzen. Mithilfe eines Schraubendrehers stellen Sie das Stellrad für die Empfangsempfindlichkeit des Akustiksensors (rechts) ein.

Hier brauchen dann nur die zwei mit dem Raspberry Pi verbundenen Motoren gegenläufig gestartet zu werden, falls der Akustiksensoren ein Geräusch wahrnimmt. Gegenläufig heißt hier, dass sich das Gestell um 180° dreht, durch Motor A in der Vorwärtsbewegung und durch Motor B in der Rückwärtsbewegung.

Im nächsten Schritt nehmen Sie den Akustiksensoren in Betrieb und schließen diesen an den Raspberry Pi an.

Schaltungsaufbau für den Akustiksensoren

Die Betriebsspannung für den Akustiksensoren beträgt laut elektronischem Datenblatt – Ergebnis der vorangegangenen Suche über eine Internet-Suchmaschine – 5 V. Dies ist auch auf der Platine bei den Steckpfosten aufgedruckt, falls hier Unsicherheit besteht. Für den Anschluss des Akustiksensors benötigen Sie weder ein Steckboard noch eine zusätzliche Schaltung, stattdessen nutzen Sie direkt die Anschlüsse auf der GPIO-Reihe.

Der Masse-Anschluss GND des Akustiksensors kommt an Pin 6 (Masse), 5 V an den 5-V-Anschluss-Pin 2, und der Zustand *out* wird in diesem Fall an GPIO7 (Pin 26) der GPIO-Leiste angeschlossen. Ist die Wiring-Pi-API installiert, dann entspricht Pin 26 in der Wiring-Pi-Zählung Nummer 11.



| Akustiksensoren-Pin | Raspberry-Pi-Bezeichnung | Raspberry-Pi-Pin | Wiring Pi |
|---------------------|--------------------------|------------------|-----------|
| + | 5 V | 2 | - |
| DO | GPIO7 | 26 | 11 |
| G | Masse | 6 | - |

Sind die drei Pins des Geräuschesensors mit der GPIO-Leiste des Raspberry Pi verbunden, dann können Sie in Sachen Programmierung starten. Egal, ob Sie per Shell-Skript, Python, C oder was auch immer die GPIO-Anschlüsse ansprechen – beachten Sie, dass für den Zugriff standardmäßig *root*-Rechte notwendig sind.

Shell-Skript für Akustiksensoren

Auf dem Weg zur Akustiksteuerung der Motoren nehmen Sie erst einmal den Sensor in Betrieb und behelfen sich mit einem einfachen Shell-Skript, was zunächst per Meldung auf dem Bildschirm darauf hinweist, ob der Sensor ein Geräusch wahrgenommen hat oder nicht. Erstellen Sie ein Projektverzeichnis (hier: *noiseswitch*) im Home-Verzeichnis des Benutzers pi.

```
sudo -i
cd /home/pi
mkdir noiseswitch
cd noiseswitch
nano noise-step1.sh
```

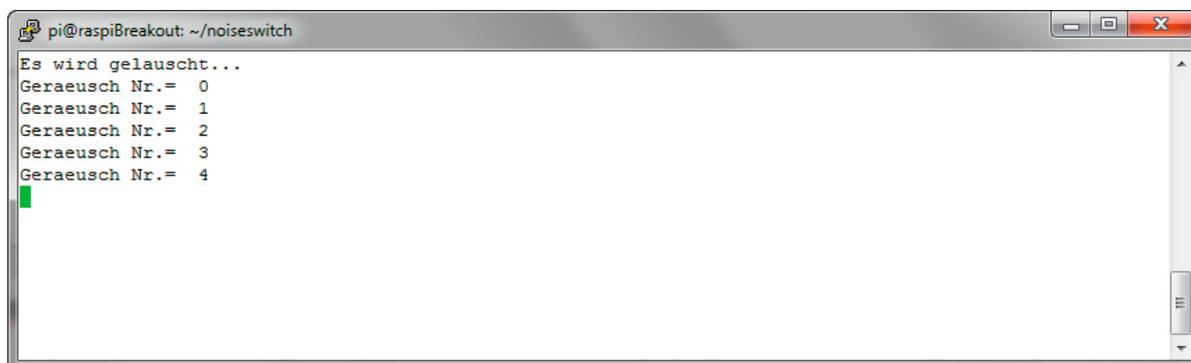
Mit dem Nano-Editor erstellen Sie ein Skript – in diesem Beispiel ist es die Datei *noise-step1.sh* – und definieren dort zunächst den genutzten Pin-Anschluss auf dem Raspberry Pi. Hier ist der Out-Ausgang des Sensors auf den Eingang GPIO7 (Pin 26) des Raspberry Pi gesteckt. Dieser Pin-Anschluss ist gleichbedeutend mit der Wiring-Pi-Nummer 11. Anschließend legen Sie mit dem *gpio mode*-Kommando die Richtung des GPIO-Pins fest. Dann geht es in die While-Schleife, in der der Zustand des Eingangs per *gpio read*-Befehl permanent abgefragt wird.

```
#!/bin/bash
#-----
# Definitionen
#-----
# Datei noise-step1.sh
# GPIO
GPIO_SOUND=7 # pin 26 - wiring pi 11 - EINGANG
#
function gpio_init(){
    gpio_export $GPIO_SOUND;
    gpio_configure_input $GPIO_SOUND;
}
#-----
function gpio_configure_output() {
    if [ -e "/sys/class/gpio/gpio${1}/direction" ]; then
        echo "out" > "/sys/class/gpio/gpio${1}/direction";
    fi
}
#-----
function gpio_configure_input() {
    if [ -e "/sys/class/gpio/gpio${1}/direction" ]; then
        echo "in" > "/sys/class/gpio/gpio${1}/direction";
    fi
}
#-----
function gpio_export() {
    echo -n "exporting gpio $1"
    if [ -e "/sys/class/gpio/gpio${1}/" ]; then
        echo "...gpio${1} already exists";
    else
        echo
        echo "${1}" > "/sys/class/gpio/export";
    fi
}
#-----
function gpio_unexport() {
    echo -n "unexporting gpio $1";
    if [ -e "/sys/class/gpio/gpio${1}/" ]; then
        echo
        echo "${1}" > "/sys/class/gpio/unexport";
    else
        echo "-> gpio${1} does not exist";
    fi
}
#-----
```



```
#-----  
function inc_zaebler(){  
  local ZAEHLERAKTUELL=$myindexnumber  
  if [ ! -z "$ZAEHLERAKTUELL" ]; then  
    myindexnumber=`echo $ZAEHLERAKTUELL+1 | bc`;  
  else  
    myindexnumber=0;  
  fi  
}  
#-----  
function zaebler_init(){  
  if [ -e "/usr/bin/bc" ]; then  
    echo -e "\n\n[sound] OK, bc bereits installiert";  
  else  
    echo -e "\n\n[sound] kein bc installiert, wird nachgeholt..";  
    sudo apt-get install bc -y;  
  fi  
}  
#-----  
# START  
#  
function init(){  
  # gpio initialisieren  
  zaebler_init;  
  gpio_init;  
}  
#-----  
# start MAIN  
init;  
clear;  
echo "[sound] Es wird gelauscht...";  
myindexnumber=0;  
# GPIO ist auf high  
# bei Geraeusch dann auf LOW - 0  
while true; do  
  if [ $(gpio -g read $GPIO_SOUND) -eq 0 ]; then  
    echo "[sound] Geraeusch Nr.= " $myindexnumber;  
    inc_zaebler;  
  fi;  
  # echo "[sound] WERT: "$(gpio -g read $GPIO_SOUND);  
  sleep 0.1;  
done;  
exit 0;  
#----- EOF -----
```

Damit der Sensor nur bei einem Geräusch anschlägt und nicht permanent eine (Dauer-)Bildschirmmeldung erzeugt, ist im nächsten Schritt die Empfindlichkeit des Sensors mit einem Elektronikschraubendreher einzustellen. Sie benötigen einen kleinen Kreuzschlitzschraubendreher, um die blaue Plastikschraube auf der Oberseite der Platine des Akustiksensors vorsichtig zu bewegen. Die Bewegung im Uhrzeigersinn erhöht die Empfindlichkeit, während die Bewegung entgegen dem Uhrzeigersinn die Sensitivität des Sensors verringert.



Kein Tinnitus beim Akustiksensoren-Betrieb: Je nach eingestellter Sensibilität erscheint nur dann eine Bildschirmmeldung, falls auch wirklich ein erkanntes Geräusch die Ursache war.

Auf Wunsch lässt sich die Logik auch in Python umsetzen, beispielsweise um den Akustiksensoren besser mit anderen Sensoren und Motoren verbinden zu können. So können Sie beispielsweise jedes Mal, wenn der Geräuschesensoren anschlägt, eine automatische Drehung einer der Motoren erfolgen lassen. **ELV**