



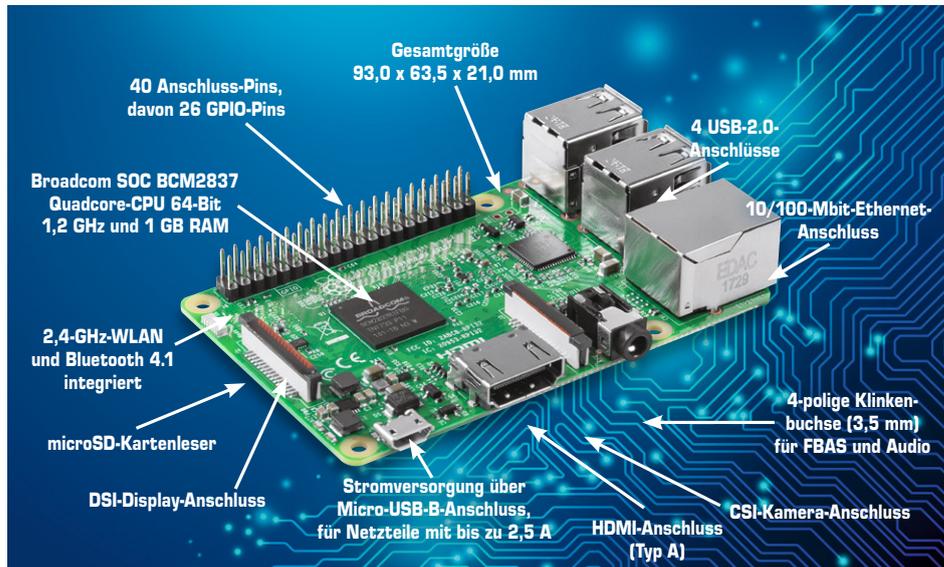
# Raspberry Pi

## Teil 11: Lage und Neigungssensor SW-520D

Artikelserie

im ELV Shop

#10036



Der Neigungssensor SW-520D ist in den einschlägigen Elektronikbörsen im Internet auch mit der Bezeichnung Tilt-Sensor oder Sensor Angle Module für kleines Geld erhältlich. Tilt-Sensoren oder allgemein elektrolitische Neigungssensoren (Inklinometer) besitzen je nach Bauweise einen Hohlkörper, in dem sich Elektroden befinden, die teilweise in einer leitfähigen Flüssigkeit (Elektrolyt) stecken. Je nach Lage des Sensors wird ein Kontakt hergestellt oder nicht. Abhängig von der Bauweise des Neigungssensors lassen sich hiermit auch die Neigungsrichtungen bestimmen. Die günstigen Modelle wie der Neigungssensor SW-520D liefern hier keine genauen Werte wie Neigungswinkel etc., sondern melden nur, ob der Sensor ausgelöst ist oder nicht.

LAST ONE

### 3.3V-12V 2 Load Control SW-520D Sensor Angle Module

Item condition: **New**

Quantity:  Last one / 3 sold in 24 hours

Price: **£5.96**

[Buy it now](#)

[Add to basket](#)

[Add to Watch list](#)

Experienced  
Seller

Free  
Postage

Last item  
available

Collect 5 Nectar points  
[Get Started](#) | [Conditions](#)

Postage: **Free** Services from outside UK | [See details](#)  
See details about international postage here. [?](#)

Item location: Hong Kong, Hong Kong  
Posts to: **Worldwide** [See exclusions](#)

Delivery: Estimated between **Tue. 11 Feb.** and **Thu. 27 Feb.** [?](#)

Payments: **PayPal** | [See payment information](#)

Returns: 14 days refund, buyer pays return postage | [Read details](#)

Protection: **ebay MONEY BACK GUARANTEE** | [See terms](#)

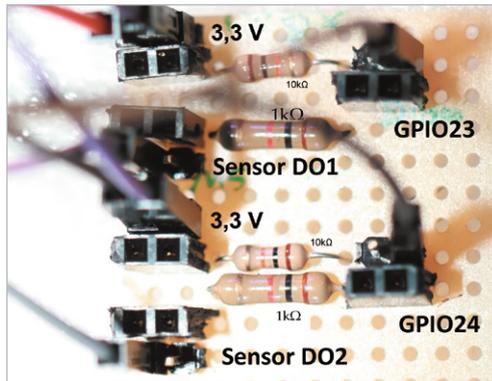
*Tilt-Sensor zum Kleinsparpreis: Je nach Zeitdruck gibt es noch günstigere Angebote, bei denen Sie allerdings ein paar Wochen Lieferzeit einkalkulieren sollten.*

Entsprechend trivial ist dann später auch die Programmierung des Sensors. Beim Einsatz des Sensors kommt es vorwiegend darauf an, diesen mechanisch so zu montieren, dass der Einsatzzweck auch perfekt zu den Anforderungen passt. So ist der Einsatz des Sensors beispielsweise an einem Garagentor im Deckenlauftorsystem denkbar: Im verschlossenen Zustand sorgt dieser dafür, dass beispielsweise das Licht ausgeschaltet bleibt. Erst wenn die Garage geöffnet wird und der Sensor ausgelöst ist, wird das Licht automatisch eingeschaltet. Oder Sie kombinieren zwei Sensoren in unterschiedlichen Richtungen: So ließe sich für kleines Geld mit dem Raspberry Pi eine elektronische Variante des Spiels Kugellabyrinth bauen.

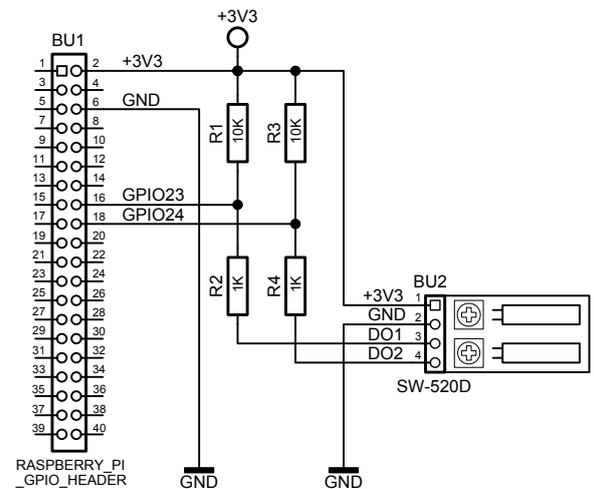


### Schaltung für den Neigungssensor

Die Schaltung für den Anschluss des Neigungssensors SW-520D an den Raspberry Pi ist abgesehen von der notwendigen Pull-up-Schaltung in wenigen Minuten aufgebaut. Da im Rahmen der Artikelreihe es immer wieder mal notwendig wurde, eine Pull-up-Schaltung aufzubauen, wurde kurzerhand die Schaltung für zwei Digitaleingänge auf einer kleinen Platine mit verschiedenen Jumper- und Steckanschlüssen realisiert.



Praktisch und schnell gelötet: Für den Anschluss eines Digitalsignals von einem Sensor oder Schalter an einem GPIO-Eingangs-Pin des Raspberry Pi ist eine Pull-up-Schaltung nötig.



Die Pull-up-Schaltung wird mit dem Widerstand 1 k $\Omega$  mit dem Digitalausgang des Neigungssensors verbunden, das andere Beinchen des Widerstands 1 k $\Omega$  mit dem Widerstand 10 k $\Omega$  und dem GPIO23-Eingang des Raspberry Pi. Das übrige Beinchen des Widerstands 10 k $\Omega$  wird mit der 3,3-V-Stromversorgung verbunden, damit das Signal am GPIO-Eingang stabil gehalten werden kann, falls vonseiten des Sensors kein Schaltsignal (Low) erfolgt.

Widerstand 10 k $\Omega$ (R1/R3)	Widerstand 1 k $\Omega$ (R2/R4)	Raspberry-Pi-Pin	Raspberry-Pi-Pin-Bezeichnung	Neigungssensor SW-520D-Pin	Neigungssensor SW-520D-Pin-Bezeichnung
2 (R3)	-	2	3,3 V	1	3,3 V
-	-	6	GND	2	GND
-	2 (R4)	-	-	4	DO2
1 (R3)	1 (R4)	18	GPIO24	-	-
2 (R1)	-	2	3,3 V	-	-
-	2 (R2)	-	-	3	DO1
1 (R1)	1 (R2)	16	GPIO23	-	-

Je zwei Widerstände 1 k $\Omega$  (R2/R4) und 10 k $\Omega$  (R1/R3) sind in einer Pull-up-Widerstandsschaltung notwendig. Damit verbinden Sie die beiden Digitalausgänge des Sensors jeweils über eine Pull-up-Schaltung mit dem Raspberry Pi, um an den als Eingang definierten GPIO-Pins ein „sauberes“ Digitalsignal zu erhalten. So bekommt es Digitalausgang DO1 bei seiner Schaltung mit den beiden Widerständen R1 (10 k $\Omega$ ) und R2 (1 k $\Omega$ ) und der zweite Ausgang DO2 mit R3 (10 k $\Omega$ ) und R4 (1 k $\Omega$ ) zu tun. Die GPIO-Eingänge werden von der Pull-up-Schaltung bestückt, die 3,3 V sorgen für ein stabiles High-Signal am entsprechenden Eingang des Raspberry Pi.

### Neigungsüberwachung mit Python

Um nun die Schaltung bzw. den Sensor in der Praxis einzusetzen, benötigen Sie eine kleine Software. Wie gewohnt legen Sie zunächst für das Projekt ein eigenes Verzeichnis im Home-Verzeichnis an.

```
cd ~
mkdir Neigungssensor_sw-520d
cd Neigungssensor_sw-520d
nano winkel-sw-520d-step1.py
```

Um nun beide Sensoren auf einer Platine abhängig voneinander gleichzeitig auszulesen, nutzen Sie einfach ein „If“-Konstrukt gepaart mit einer „And“-Bedingung. Diese befindet sich innerhalb einer dauerhaften „While“-Schleife, damit die Zustandsänderungen permanent geprüft und übermittelt werden. Diese unterbrechen Sie mit der Tastenkombination <STRG><C>, um das Testprogramm für den Neigungssensor zu beenden.

```
# -*- coding: utf-8 -*-
#!/usr/bin/python
# -----
# winkel-sw-520d-step1.py in der Schleife
import time
import RPi.GPIO as GPIO
import os
# -----
GPIO.setmode(GPIO.BCM)
# *****
```



```
# Definition der Pins. Hier Pin 16 = GPIO23
#                               Pin 18 = GPIO24
#
GPIO_s0 = 23 # S0/GPIO23/16/4 # Modul 1 - sensor 1
GPIO_s1 = 24 # S1/GPIO24/18/5 # Modul 1 - sensor 2
def setup():
    GPIO.setup(GPIO_s0,GPIO.IN)
    GPIO.setup(GPIO_s1,GPIO.IN)
    return
# Jeder Input mit Pullup Schaltung !!!
# -----
# Start Python-Skript
# -----
os.system('clear')
setup()
print("[SW-520D] Messung ...")
try:
    while True:
        print("Sensor 1.....")
        print("s0=", GPIO.input(GPIO_s0), "s1=", GPIO.input(GPIO_s1))
        if(GPIO.input(GPIO_s0) and (GPIO.input(GPIO_s1))):
            print("Beide LOW")
        elif (GPIO.input(GPIO_s0) and not (GPIO.input(GPIO_s1))):
            print("s0 LOW - s1 HIGH")
        elif not (GPIO.input(GPIO_s0)) and (GPIO.input(GPIO_s1)):
            print("s0 HIGH - s1 LOW")
        elif not (GPIO.input(GPIO_s0)) and not (GPIO.input(GPIO_s1)):
            print("s0 HIGH - s1 HIGH")
        time.sleep(2) # Sekunden warten
except KeyboardInterrupt:
    # CTRL-C gedrueckt Reset GPIO
    print("[SW-520D] Messung abgebrochen.")
    GPIO.cleanup()
# ----- EOF -----
```

Auf dem Bildschirm stellt sich die Funktionsweise der dargestellten Python-Lösung wie folgt dar – je nach Position des Sensors liefert dieser seinen Status.

```
pi@raspiBreakout: ~/winkel-sw-520d
[SW-520D] Messung ...
Sensor 1.....
s0= 1 s1= 1
Beide LOW
Sensor 1.....
s0= 1 s1= 1
Beide LOW
Sensor 1.....
s0= 1 s1= 1
Beide LOW
Sensor 1.....
s0= 0 s1= 0
s0 HIGH - s1 HIGH
Sensor 1.....
s0= 0 s1= 0
s0 HIGH - s1 HIGH
Sensor 1.....
s0= 1 s1= 0
s0 LOW - s1 HIGH
Sensor 1.....
s0= 1 s1= 0
s0 LOW - s1 HIGH
Sensor 1.....
s0= 0 s1= 0
s0 HIGH - s1 HIGH
```

*Je nach Lage des Neigungssensors schlagen die beiden Sensoren unterschiedlich an. Der Status wird hier einfach mit HIGH und LOW ausgegeben.*

Ein Tilt-/Neigungssensor ist für einfache Anwendungen unheimlich praktisch, sollen hingegen der Neigungswinkel und andere Messwerte genauer dargestellt werden, dann stößt ein solcher Sensor an seine (physikalischen) Grenzen. Für diesen Fall nutzen Sie Platinen bzw. Sensoren, die mit einem Gyrometer ausgestattet sind. **ELV**