

www.scoutlab.de/kit



## Morsen

# Morsegerät bauen

#### Material

- 1x Steckbrett
- 1x Buzzer
- 1x Taster
- 4x Kabel



Abbildung: Rückseite Buzzer

**Scratch:** morsen.sb in deinem Projekt-Ordner für Scratch. **Python:** morsen.py in deinem Projekt-Ordner für Python.

#### Aufbau

Wir erweitern die Schaltung aus dem Schritt "Taster auslesen" um einen Buzzer. Bitte beachte, dass du den Buzzer

richtig anschließt. Stecke den Pin mit "-" in die obere Leiste auf deinem Breadboard. In der Zeichnung ist dieser Pin schwarz markiert.

## Scratch-Programm

```
wenn was a piocentrick and alle sende config2lout an alle sende config2lout an alle sende config2lout an alle wiederhole fortland falls went von Sensor gpio18 = 0

sende beep an alle und warte sonst sende gpio21off an alle warte 0.01 Sek.

sende gpio21off an alle warte 0.01 Sek.
```

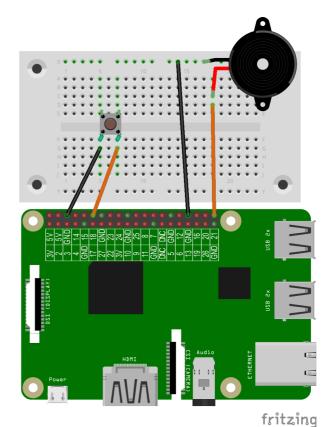


Abbildung: Morsegerät bauen



verband Christlicher
PFADFINDER UND
PFADFINDER

www.scoutlab.de/kit

Der Block **gpioserveron** startet den notwendigen GPIO-Server, damit das Programm auf die GPIO-Pins zugreifen kann. Mit **config18in** wird der GPIO-Pin für den Taster als Eingang, mit **config21out** der GPIO-Pin 21für den Buzzer als Ausgang definiert. Eine fortlaufend wiederholende Schleife prüft, ob der Taster (**Wert von Sensor gpio18=0**) gedrückt wird. Wenn die Bedingung erfüllt ist, wird der **Programmblock beep** aufgerufen. Sonst schaltet **gpio21off** den Buzzer aus.

Wissen: Eine Buzzer (Piezo-Lautsprecher) kann technisch nur einen kurzen Ton abgeben und ist für das Morsen ungeeignet. Um das auszugleichen hab wir einen eigene Funktion, oder Programmblock erstellt. Der Programmblock schaltet in einer Endlosschleife den Buzzer für eine 1/100 Sekunde ein, schaltet den Buzzer wieder für 1/100 Sekunde wieder an.

Für das menschliche Ohr hört sich das nach einem Dauerton an. Damit ist das Summen jetzt so lange, wie der Taster gedrückt wird. Damit können beliebige Morse-Codes gesendet werden.

VCP VERBAND CHRISTLICHER PFADFINDER UND PFADFINDER

www.scoutlab.de/kit

## **Python-Programm**

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
from time import sleep
TasterPin = 18
BuzzerPin = 21
GPIO.setup(BuzzerPin, GPIO.OUT)
GPIO.output(BuzzerPin, GPIO.LOW)
GPIO.setup(TasterPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
def beep():
    GPIO.output(BuzzerPin,GPIO.HIGH)
    sleep(0.01)
    GPIO.output(BuzzerPin,GPIO.LOW)
    sleep(0.01)
while True:
  if GPIO.input(TasterPin) == False:
    beep()
  else:
    GPIO.output(BuzzerPin,GPIO.LOW)
sleep(0.2)
```



VCP VERBAND CHRISTLICHER PFADFINDER UND PFADFINDER

www.scoutlab.de/kit

Buchstabe	Morse-Code
A	. –
В	
С	
D	
E	
F	
G	
Н	
1	
J	
K	
L	
М	
N	
0	
Р	
Q	



VCP VERBAND CHRISTLICHER PFADFINDER UND PFADFINDER

www.scoutlab.de/kit

Buchstabe	Morse-Code
R	. – .
S	
Т	-
U	
V	
W	
X	
Υ	
Z	



VCP VERBAND CHRISTLICHER PFADFINDER UND PFADFINDER

www.scoutlab.de/kit

Zahl	Morse-Code
1	
2	
3	
4	
5	
6	<b></b>
7	
8	
9	
0	



www.scoutlab.de/kit



#### Text in Morse-Code übersetzen

MorseAlphabet = { "A" : ".-", "B" : "-...", "C" : "-.-.", "D" : "-..", "E":".",
"F":".-.", "G" : "--.", "H" : "....", "l" : "..", "J" : ".---", "K" : "-.-", "L": ".-..", "M": "--", "N" : "-.", "O" : "---". "P" : ".--.", "Q": "--.-", "R" : ".-.", "S":"...", "T" : "-", "U" : "..-", "V" : "...-" "W" : ".--" "X": "-..-", "Y" : "-.--", "Z" : "--..", "∠ . "":"", "1" : ".----", "2" : "..---", "3" : "...--", "4":"....-", "5" : ".....", "6" : "-....", "7" : "--...", "8" : "---.." "9" : "----.", "0" : "-----", ".": ".-.-.", ",":"--..-", ":" : "---...", "?":"..-.." "" : ".---.", "-":"-....-", "/" : "-..-.", "@":".--.-.", msg = raw\_input('Nachricht: ')

for char in msg:

print (morseAlphabet[char.upper()]),

Python: texteingabe-in-morsecode.py in deinem Projekt-Ordner für Python. Bitte beachten: das Beispiel funktioniert nur mit Python2