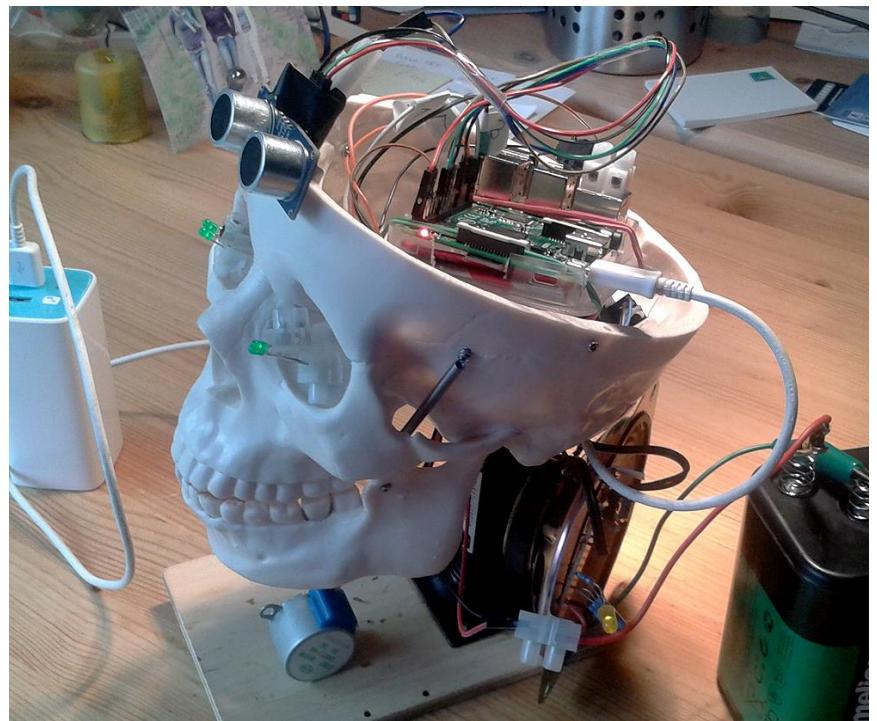


# Geisterbahn mit Raspberry

Eine Zusammenschaltung aus Raspberry, Ultraschallsensor, LEDs, Schrittmotor und Verstärker mit Lautsprecher ermöglicht einen Beitrag zur Halloween-Party oder zur Schulfest-Geisterbahn!



```
#!/usr/bin/python
import time, RPi.GPIO as GPIO, subprocess,os,from time import sleep; GPIO.setmode(GPIO.BCM); GPIO.setwarnings(False)
# Konstanten fuer GPIO-Pins und andere Konstanten definieren:
A=19; B=16; C=26; D=20; TRIGGER = 17; ECHO = 22; LED = 14; dtim = 0.001; t05 = 0.5; Schallgeschwindigkeit = 331
# Schritte 1 - 8 definieren fuer Schrittmotor
def Step1():
    GPIO.output(D, True); sleep (dtim); GPIO.output(D, False)
def Step2():
    GPIO.output(D, True); GPIO.output(C, True); sleep (dtim); GPIO.output(D, False); GPIO.output(C, False)
def Step3():
    GPIO.output(C, True); sleep (dtim); GPIO.output(C, False)
def Step4():
    GPIO.output(B, True); GPIO.output(C, True); sleep (dtim); GPIO.output(B, False); GPIO.output(C, False)
def Step5():
    GPIO.output(B, True); sleep (dtim); GPIO.output(B, False)
def Step6():
    GPIO.output(A, True); GPIO.output(B, True); sleep (dtim); GPIO.output(A, False); GPIO.output(B, False)
def Step7():
    GPIO.output(A, True); sleep (dtim);GPIO.output(A, False)
def Step8():
    GPIO.output(D, True); GPIO.output(A, True); sleep (dtim); GPIO.output(D, False); GPIO.output(A, False)
# Ständig durchlaufende Schleife:
while True:
    GPIO.setup(TRIGGER,GPIO.OUT); GPIO.setup(ECHO,GPIO.IN); GPIO.setup(7, GPIO.OUT); GPIO.setup(LED, GPIO.OUT); GPIO.output(LED, False)
    GPIO.setup(A,GPIO.OUT); GPIO.setup(B,GPIO.OUT); GPIO.setup(C,GPIO.OUT); GPIO.setup(D,GPIO.OUT)
    GPIO.output(A, False); GPIO.output(B, False); GPIO.output(C, False); GPIO.output(D, False)
    # Messung auslösen
    GPIO.output(TRIGGER, False); time.sleep(0.5); GPIO.output(TRIGGER, True); time.sleep(0.00001); GPIO.output(TRIGGER, False)
    while GPIO.input(ECHO) == 0:
        pass
    # Startzeit festhalten
    Start = time.time()
    while GPIO.input(ECHO) == 1:
        pass
    # Stopzeit festhalten:
    Stop = time.time()
    # Zeitdifferenz berechnen:
    Zeit = Stop - Start
    # Entfernung in Meter berechnen und anzeigen:
    Entfernung = Zeit * (Schallgeschwindigkeit/2); Entfernung = round(Entfernung, 2); print "Abstand: ", Entfernung, " m"
    # Reaktion auf Annäherung festlegen:
    if Entfernung < 0.5:
        GPIO.output(LED, True)
        # 1/4 Umdrehung
        for i in range (380):
            Step8(); Step7(); Step6(); Step5(); Step4(); Step3(); Step2(); Step1()
            a = subprocess.call(["sudo", "omxplayer","--vol","-1000", "-o", "local","skull.wav"])
        for i in range (380):
            Step1(); Step2(); Step3(); Step4(); Step5(); Step6(); Step7(); Step8()
        GPIO.output(LED, False)
    GPIO.cleanup()
```