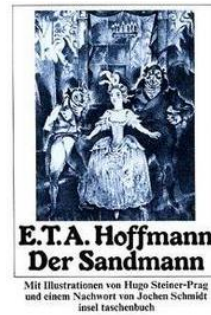


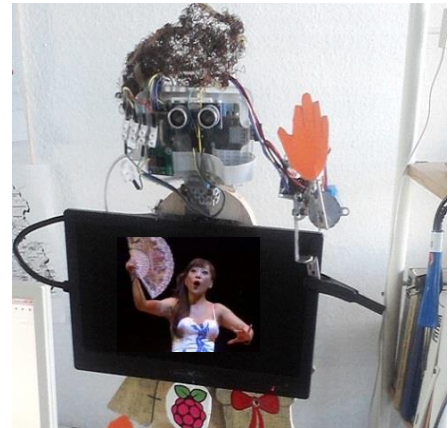


Olimpia-Pi



In E.T.A. Hoffmanns Erzählung "Der Sandmann" verfällt der Student Nathanael einer mechanischen Puppe namens Olimpia. Diese hat der wahnsinnigen Professor Spalanzani konstruiert und gibt sie nun als seine Tochter aus.

Der aktuelle Stand der Sensor-Aktor-Mikrocontroller-Technologie macht es ambitionierten Bastlern mittlerweile möglich, mit Ultraschallsensoren, Mikrofonen, integrierten Videokameras, Displays, Lautsprechern, Schritt- und Servomotoren, Mikropumpen etc. elektromechanische Maschinen für knapp 100 € zu erschaffen, die der Idee einer Olimpia gruselig nahe kommen. Und diese „Androiden“ machen in handgreiflich-praktischem Sinne erlebbar, dass sie softwareseitig relativ einfach manipulierbar sind.



Das hier gezeigte Exemplar ist bewusst einfach und roh gehalten: ein Raspberry-Py-Einplatinen-Computer mit Videokamera, ein Ultraschallsensor-Modul, ein HDMI-Bildschirm, ein Schrittmotor. Die übergeordnete Steuerung erfolgt über das Programm „olimpia.py“ (siehe unten), das bei Bedarf weitere Programme, z.B. zum Abspielen von Multimedia-Dateien, auf Betriebssystemebene aufruft.

Für die Einspielung des Ausschnitts aus dem „Lied der Olymppia“ - das sich im zweiten Akt in der Oper „Hoffmanns Erzählungen“ von Jaques Offenbach findet - haben wir auf ein Youtube-Video zurückgegriffen, das die Sängerin Sumi Jo in der entsprechenden Rolle zeigt.

Das Programm „olimpia.py“:

```
#!/usr/bin/python
from time import sleep;import time, picamera,RPi.GPIO as GPIO,os,subprocess
GPIO.setmode(GPIO.BOARD); GPIO.setwarnings(False) #GPIO Moduswahl und Pinbenennungen
TRIGGER = 16; ECHO = 18; BUTTON=26 #GPIO Pinbenennungen
# Fuer Schrittmotor verwendete Pins am Raspberry Pi, Zeitintervalle, Schallgeschwindigkeit in m/s:
A=3; B=5; C=7; D=11; dtime = 0.005; t05 = 0.5; Schallgeschwindigkeit = 331
# Pins aus Ausgange definieren, Werte setzen:
GPIO.setup(BUTTON, GPIO.IN); 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)
# Motor-Schritte 1 - 8 festlegen:
def Step1():
    GPIO.output(D, True); sleep(dtime); GPIO.output(D, False)
def Step2():
    GPIO.output(D, True); GPIO.output(C, True); sleep(dtime); GPIO.output(D, False); GPIO.output(C, False)
def Step3():
    GPIO.output(C, True); sleep(dtime); GPIO.output(C, False)
def Step4():
    GPIO.output(B, True); GPIO.output(C, True); sleep(dtime); GPIO.output(B, False); GPIO.output(C, False)
def Step5():
    GPIO.output(B, True); sleep(dtime); GPIO.output(B, False)
def Step6():
    GPIO.output(A, True); GPIO.output(B, True); sleep(dtime); GPIO.output(A, False); GPIO.output(B, False)
def Step7():
    GPIO.output(A, True); sleep(dtime); GPIO.output(A, False)
def Step8():
    GPIO.output(D, True); GPIO.output(A, True); sleep(dtime); GPIO.output(D, False); GPIO.output(A, False)
# Start der Dauerschleife mit Entfernungsmessung und Reaktionen:
while True:
    GPIO.setup(TRIGGER, GPIO.OUT); GPIO.setup(ECHO, GPIO.IN) # Messung ausloesen
    GPIO.output(TRIGGER, False); time.sleep(0.1); GPIO.output(TRIGGER, True)
    time.sleep(0.00001); GPIO.output(TRIGGER, False)
    while GPIO.input(ECHO) == 0:
        pass
    Start = time.time() # Startzeit festhalten
    while GPIO.input(ECHO) == 1: # Stopzeit festhalten
        pass
    Stop = time.time() # Zeitdifferenz berechnen
    Zeit = Stop - Start
    # Entfernung in Meter berechnen - Reaktionen definieren:
    Entfernung = Zeit * (Schallgeschwindigkeit/2)
    if Entfernung > 1.2 and Entfernung < 2.0:
        a = subprocess.call(["sudo", "omxplayer", "-o", "hdmi", "olimpia.mp4"])
    if Entfernung < 0.7 and Entfernung > 0.3:
        p= subprocess.Popen("sh vid.sh", shell=True)
        a = subprocess.call(["sudo", "omxplayer", "-o", "hdmi", "ich-seh-dich.wav"])
    if Entfernung < 0.3:
        a = subprocess.call(["sudo", "omxplayer", "-o", "hdmi", "s.mp4"])
        for i in range(30):
            Step1(); Step2(); Step3(); Step4(); Step5(); Step6(); Step7(); Step8()
        for i in range(29):
            Step7(); Step6(); Step5(); Step4(); Step3(); Step2(); Step1()
```