Wasserstandsüberwachung mit Raspberry Pi 4 und HC-SR04 Ultraschallsensor

Projektbericht

Autor: [Dein Name]

Datum: [Datum]

# Inhaltsverzeichnis

1. 1. Einleitung
2. 2. Hardware-Komponenten
3. 3. Software-Code
4. 3.1. Grundkonfiguration
5. 3.2. Modulkonfiguration
6. 3.3. Code zur Messung des Wasserstands
7. 3.4. Code zur Steuerung der Philips Hue Steckdose
8. 3.5. Integrierter Code zur Überwachung und Steuerung
9. 4. Hardware-Liste
10. 5. Blockschaltbild
11. 6. Projekt-Ablauf
12. 7. Fazit

# 1. Einleitung

Dieses Projekt beschreibt die Realisierung einer Wasserstandsüberwachung für einen Eimer mittels eines Raspberry Pi 4 und eines HC-SR04 Ultraschallsensors. Sobald der Wasserstand 95% erreicht, wird eine Philips Hue Steckdose angesteuert, die eine Pumpe aktiviert, um den Eimer zu entleeren.

# 2. Hardware-Komponenten

- Raspberry Pi 4  
- HC-SR04 Ultraschallsensor  
- Philips Hue Steckdose  
- Wasserpumpe  
- Eimer  
- Diverse Kabel  
- Steckbrett (Breadboard)

![Hardware-Komponenten](./bilder/hardware.jpg)

# 3. Software-Code

## Grundkonfiguration

Der Raspberry Pi wird mit dem Betriebssystem Raspbian ausgestattet und alle notwendigen Pakete werden installiert.  
```bash  
sudo apt update  
sudo apt install python3 python3-pip  
pip3 install phue  
```

## Modulkonfiguration

Der HC-SR04 Sensor wird über die GPIO Pins des Raspberry Pi ausgelesen. Hierzu wird eine Python-Bibliothek verwendet, um die Entfernungsdaten zu messen und den Wasserstand zu berechnen.

### Code zur Messung des Wasserstands

```python  
import RPi.GPIO as GPIO  
import time  
  
# Pin-Definitionen  
TRIG = 23  
ECHO = 24  
  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(TRIG, GPIO.OUT)  
GPIO.setup(ECHO, GPIO.IN)  
  
def get\_distance():  
 GPIO.output(TRIG, True)  
 time.sleep(0.00001)  
 GPIO.output(TRIG, False)  
  
 while GPIO.input(ECHO) == 0:  
 start\_time = time.time()  
  
 while GPIO.input(ECHO) == 1:  
 end\_time = time.time()  
  
 duration = end\_time - start\_time  
 distance = (duration \* 34300) / 2  
  
 return distance  
  
try:  
 while True:  
 distance = get\_distance()  
 print(f"Distance: {distance:.2f} cm")  
 time.sleep(1)  
except KeyboardInterrupt:  
 GPIO.cleanup()  
```

### Code zur Steuerung der Philips Hue Steckdose

```python  
from phue import Bridge  
  
bridge\_ip = 'IP\_ADDRESS\_OF\_BRIDGE'  
b = Bridge(bridge\_ip)  
b.connect()  
  
def control\_pump(turn\_on):  
 lights = b.get\_light\_objects('name')  
 pump\_light = lights['PumpSocket']  
  
 if turn\_on:  
 pump\_light.on = True  
 else:  
 pump\_light.on = False  
  
# Beispiel zur Steuerung der Pumpe  
control\_pump(True) # Pumpe einschalten  
control\_pump(False) # Pumpe ausschalten  
```

### Integrierter Code zur Überwachung und Steuerung

```python  
import RPi.GPIO as GPIO  
import time  
from phue import Bridge  
  
# Pin-Definitionen und Philips Hue Bridge  
TRIG = 23  
ECHO = 24  
bridge\_ip = 'IP\_ADDRESS\_OF\_BRIDGE'  
b = Bridge(bridge\_ip)  
b.connect()  
  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(TRIG, GPIO.OUT)  
GPIO.setup(ECHO, GPIO.IN)  
  
def get\_distance():  
 GPIO.output(TRIG, True)  
 time.sleep(0.00001)  
 GPIO.output(TRIG, False)  
  
 while GPIO.input(ECHO) == 0:  
 start\_time = time.time()  
  
 while GPIO.input(ECHO) == 1:  
 end\_time = time.time()  
  
 duration = end\_time - start\_time  
 distance = (duration \* 34300) / 2  
 return distance  
  
def control\_pump(turn\_on):  
 lights = b.get\_light\_objects('name')  
 pump\_light = lights['PumpSocket']  
  
 if turn\_on:  
 pump\_light.on = True  
 else:  
 pump\_light.on = False  
  
try:  
 while True:  
 distance = get\_distance()  
 water\_level\_percentage = (MAX\_WATER\_LEVEL - distance) / MAX\_WATER\_LEVEL \* 100  
  
 if water\_level\_percentage >= 95:  
 control\_pump(True)  
 else:  
 control\_pump(False)  
  
 print(f"Water Level: {water\_level\_percentage:.2f}%")  
 time.sleep(1)  
except KeyboardInterrupt:  
 GPIO.cleanup()  
```

# 4. Hardware-Liste

- 1x Raspberry Pi 4  
- 1x HC-SR04 Ultraschallsensor  
- 1x Philips Hue Steckdose  
- 1x Wasserpumpe  
- 1x Eimer  
- 1x Steckbrett (Breadboard)  
- Verbindungskabel

# 5. Blockschaltbild

![Blockschaltbild](./bilder/blockschaltbild.jpg)

# 6. Projekt-Ablauf

1. Aufbau der Hardware:  
 - Verkabelung des HC-SR04 Sensors mit dem Raspberry Pi.  
 - Verbindung der Wasserpumpe mit der Philips Hue Steckdose.  
  
2. Software-Installation:  
 - Installation des Betriebssystems auf dem Raspberry Pi.  
 - Installation der notwendigen Python-Pakete.  
  
3. Programmierung:  
 - Schreiben und Testen des Codes zur Messung des Wasserstands.  
 - Implementierung der Steuerung der Philips Hue Steckdose.  
  
4. Integration und Test:  
 - Zusammenführen des Codes.  
 - Testen des gesamten Systems im realen Szenario.

# 7. Fazit

Das Projekt zeigt eine einfache Methode zur Überwachung des Wasserstands und automatisierten Steuerung einer Pumpe mittels eines Raspberry Pi und eines HC-SR04 Ultraschallsensors. Durch die Nutzung der Philips Hue Steckdose kann die Steuerung bequem über das bestehende Smart-Home-System erfolgen. Das System ist zuverlässig und kann bei Bedarf erweitert werden.