

# Inbetriebnahmeprotokoll

## 1. Beteiligte

Beteiligt an der Inbetriebnahme waren Daniel Kühnel und Max Ullmann.

## 2. Datum der Inbetriebnahme

Die Inbetriebnahme fand am 30.11.2022 um 09:30 Uhr statt.

## 3. Aktive Geräte

Zu dem Zeitpunkt wurde der Raspberry Pi4 in Betrieb genommen. Der DHT11-Sensor wurde anschließend in Betrieb genommen und dieser getestet. Gemessen wurde die aktuelle Raumtemperatur von ca. 23 °C. Zum Anzeigen der Temperatur wurde zusätzlich die Siebensegmentanzeige in Betrieb genommen.

## 4. Prüfung der Funktionalität

Der Quelltext spricht den DHT11 - Sensor an und liest den aktuellen Temperaturwert und Feuchtigkeitswert aus. Dieser wird in der Konsole ausgegeben und auf Plausibilität überprüft. Die Temperatur wird in einer Schleife 20-mal ausgelesen und ausgegeben. Dieser Versuch verläuft in einem Zeitraum von 5 Minuten, dank einer Verzögerung von 15 Sekunden pro Durchlauf.

```
GNU nano 5.4
import RPi.GPIO as GPIO
import dht11
import time

# initialize GPIO

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()

z = 1
i = 1
while i<21 :

    # read data using pin 14
    instance = dht11.DHT11(pin = 4)
    result = instance.read()

    while not result.is_valid():
        result = instance.read()
    print ("")
    print ("Versuch: " + str(z))
    temp = round(result.temperature)
    humi = round(result.humidity)
    print ("Temperature: %-3.0fc" % result.temperature)
    print ("Humidity: %-3.0f%%" % result.humidity)
    time.sleep(15)
    i = i + 1
    z = z + 1
```

Abbildung 1 - Quelltext DHT11 Funktionalitätstest

Die Temperatur wurde mittels Auflegens eines Fingers auf den Sensor manipuliert. Diese Änderungen der Umgebung des Sensors wird ab Versuch 11 durchgeführt. Zu sehen ist, dass die ausgegebenen Werte eine signifikante Änderung aufweisen, besonders im Bereich der Luftfeuchte.

```
Versuch: 1  
Temperature: 23 C  
Humidity: 41 %  
  
Versuch: 2  
Temperature: 23 C  
Humidity: 41 %  
  
Versuch: 3  
Temperature: 23 C  
Humidity: 40 %  
  
Versuch: 4  
Temperature: 23 C  
Humidity: 40 %  
  
Versuch: 5  
Temperature: 23 C  
Humidity: 41 %  
  
Versuch: 6  
Temperature: 23 C  
Humidity: 40 %  
  
Versuch: 7  
Temperature: 23 C  
Humidity: 40 %  
  
Versuch: 8  
Temperature: 23 C  
Humidity: 40 %  
  
Versuch: 9  
Temperature: 23 C  
Humidity: 40 %  
  
Versuch: 10  
Temperature: 23 C  
Humidity: 41 %
```

*Abbildung 2 -  
Funktionalitätstest  
DHT11 (11-20)*

```
Versuch: 11  
Temperature: 23 C  
Humidity: 41 %  
  
Versuch: 12  
Temperature: 25 C  
Humidity: 95 %  
  
Versuch: 13  
Temperature: 25 C  
Humidity: 95 %  
  
Versuch: 14  
Temperature: 25 C  
Humidity: 95 %  
  
Versuch: 15  
Temperature: 26 C  
Humidity: 95 %  
  
Versuch: 16  
Temperature: 26 C  
Humidity: 95 %  
  
Versuch: 17  
Temperature: 26 C  
Humidity: 95 %  
  
Versuch: 18  
Temperature: 26 C  
Humidity: 95 %  
  
Versuch: 19  
Temperature: 26 C  
Humidity: 95 %  
  
Versuch: 20  
Temperature: 26 C  
Humidity: 95 %
```

*Abbildung 3 -  
Funktionalitätstest  
DHT11 (1-10)*

Das Skript kann Anlage 2 der Hauptdokumentation entnommen werden.

Die Siebensegmentanzeige wurde mit dem folgenden Quelltext in Betrieb genommen und zum Anzeigen der Temperatur (links neben dem Doppelpunkt) und der Luftfeuchtigkeit (rechts neben dem Doppelpunkt) genutzt:

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  import board
4  import RPi.GPIO as GPIO
5  import dht11
6  import time
7
8  from adafruit_ht16k33.segments import Seg7x4
9
10 GPIO.setwarnings(False)
11 GPIO.setmode(GPIO.BCM)
12 GPIO.cleanup()
13 i2c = board.I2C()
14 segment = Seg7x4(i2c, address=0x70)
15 #segment der I2C Adresse 0x70 und die Displaydefinition zuweisen
16 segment.fill(0)
17 # Initialisierung des Displays.
18 # Muss einmal ausgeführt werden bevor das Display benutzt wird.
19 print ("STRG+C Druecken zum beenden.")
20 #print Befehl für Ausgabe zum beenden des Scriptes
21 #Schleife welche dauerhaft die Zeit updated und sie auf dem Display anzeigt.
22 try:
23     while(True):
24         #segment.fill(0)
25
26         instance = dht11.DHT11(pin = 4)
27         result = instance.read()
28
29         while not result.is_valid():
30             result = instance.read()
31
32         temp = int(round(result.temperature,0))
33         humi = int(round(result.humidity,0))
34
35         # Doppelpunkt zwischen Nummern
36         segment.colon = True
37
38         # Temperatur
39         segment[0] = str(int(temp / 10)) # Zehnerzahlen
40         segment[1] = str(temp % 10) # Einerzahlen
41         # Luftfeuchtigkeit
42         segment[2] = str(int(humi / 10)) # Zehnerzahlen
43         segment[3] = str(humi % 10) # Einerzahlen
44
45         segment.show() # Wird benötigt um die Display LEDs zu updaten.
46         time.sleep(1) # Warte eine Sekunde
47 except KeyboardInterrupt:
48     segment.fill(0)
49
```

Abbildung 4 - Funktionalitätstest Siebensegmentanzeige

Das Skript kann Anlage 3 der Hauptdokumentation entnommen werden.

## 5. Ergebnis

Das Ergebnis zeigt, dass der Sensor ordnungsgemäß funktioniert und auf Veränderungen der Umgebung, wie erwartet, reagiert. Die Siebensegmentanzeige zeigt, in nahezu Echtzeit, die aktuelle Temperatur und Luftfeuchtigkeit an.