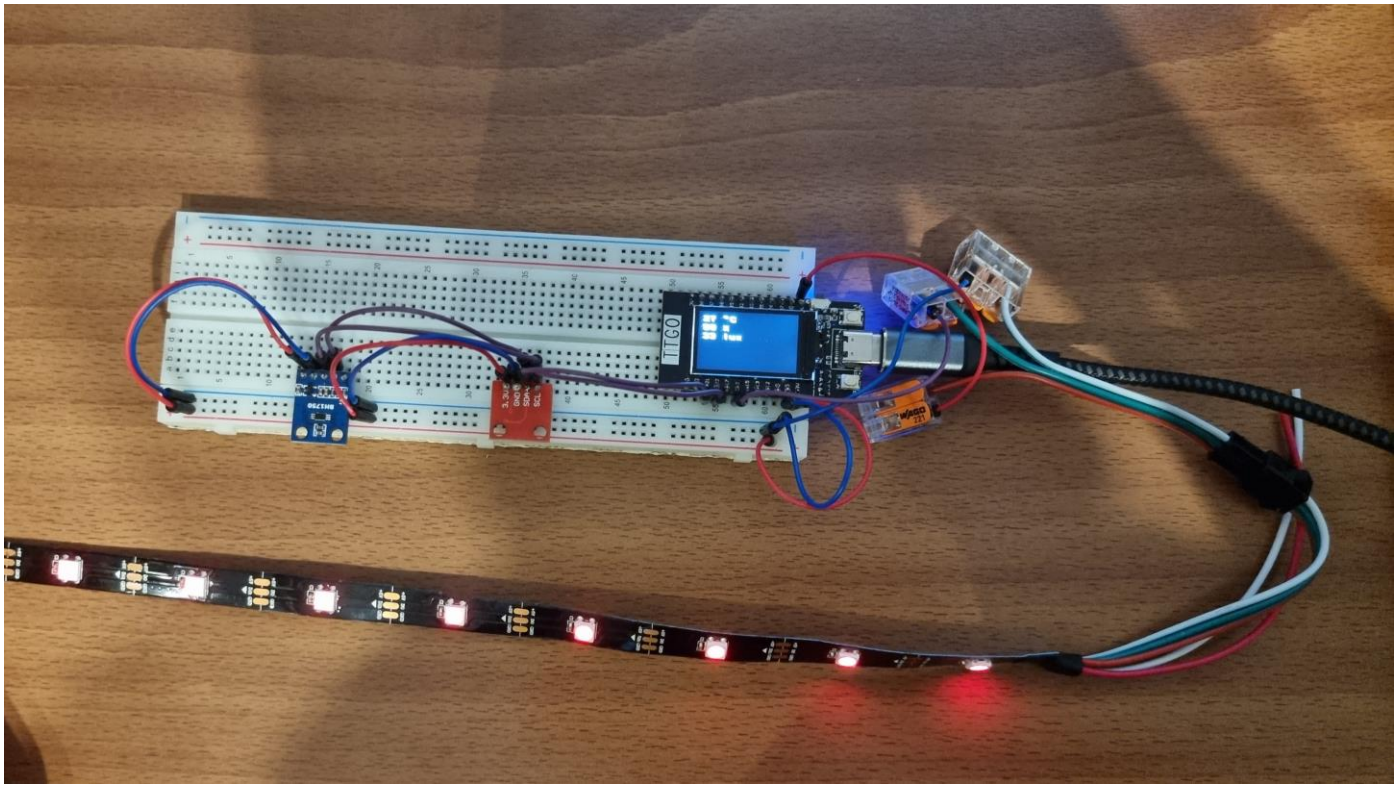
WETTERSTATION

MICROCONTROLLER PROJEKT



Matthias Grafe
ETS2021
20.06.2022

Inhaltsverzeichnis

WETTERSTATION

Kundenauftrag

Die Meyer Holz GmbH hat ein modernes Designgehäuse aus Holz erstellt und möchte in das Gehäuse eine Wetterstation integrieren. Das Gehäuse soll in Massenproduktion gehen und an den Kunden verkauft werden.

Die MG GmbH bekommt den Auftrag die Wetterstation zu planen und zu erstellen.

Realisierung

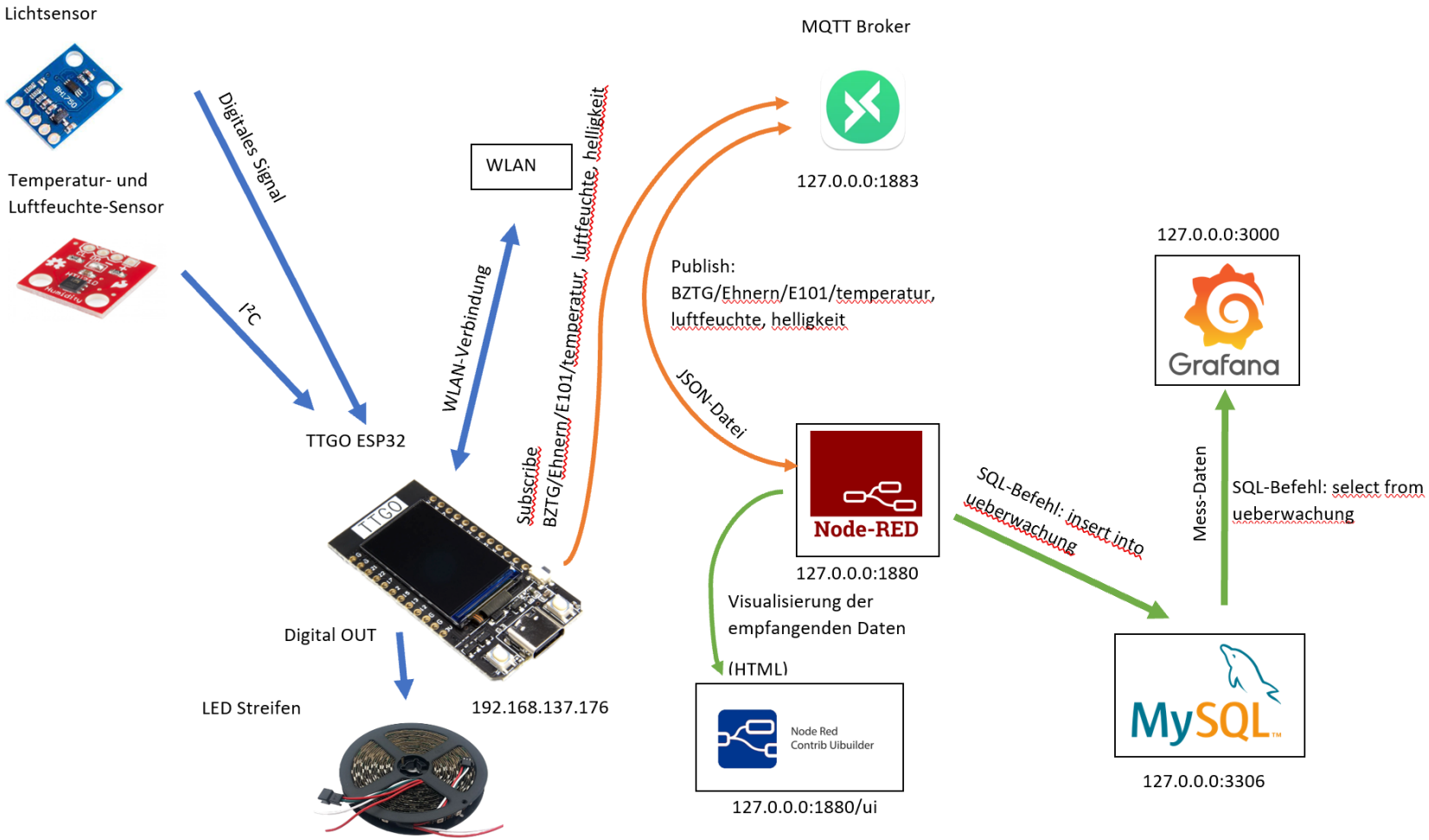
Luftfeuchte und Temperatur sollen für ein Display (ESP32 OLED) gemessen werden.
Zusätzlich soll die Luftfeuchte in drei Abstufungen mit LED´s angezeigt werden. Rot für schlecht, gelb für grenzwertig und grün für gut.
Ein BH1750 Sensor misst die Umgebungshelligkeit und je nach Lichteinfall, sollen die LED´s mit voller Stärke oder gedimmt leuchten.

Verwendete Hardware:

- Microcontroller: ESP32 (TTGO)
- Sensoren:
  - HTU2X, I2C Temperatur- und Luftfeuchtigkeits-Sensor
  - BH1750, Helligkeitssensor
- Aktoren:
  - LED Stripe (WS2812)
  - Display (ESP32)

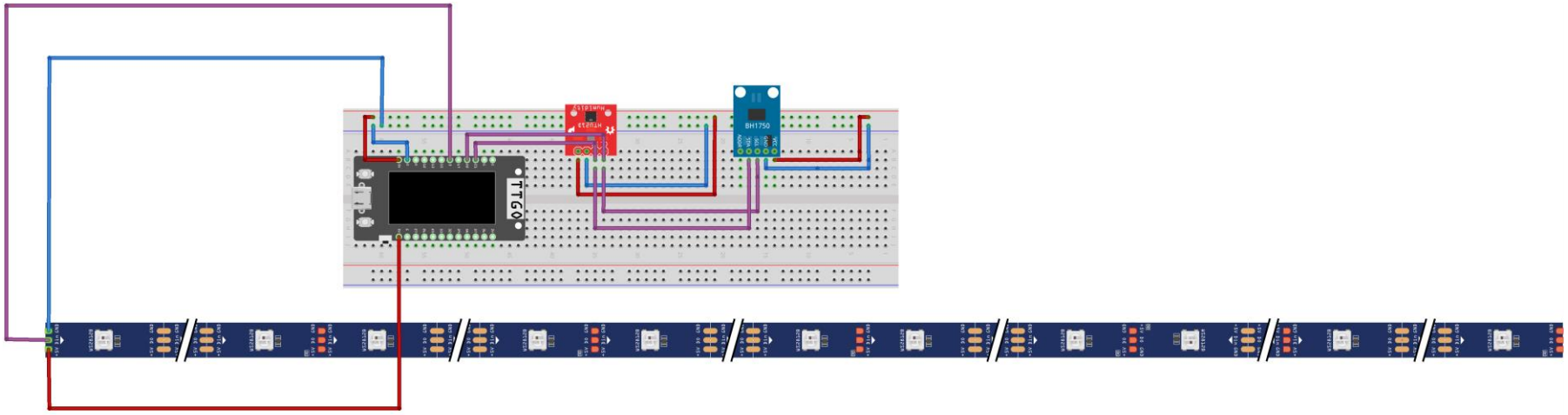
WETTERSTATION

Blockschaltbild



WETTERSTATION

Steckplan



Code

hauptprogramm.py > ...

```
# Matthias Grafe
# ETS2021
# µController Projekt
# 20.06.2022
# Versionsnummer 2.5 vom 20.06.2022


"""
Was macht das Programm?
Ein HTU2X Sensor misst die Temperatur und Luftfeuchtigkeit. Ein BH1750 Helligkeitssensor misst die Helligkeit.
Die drei Werte werden auf dem TFT Display vom ESP32 ausgegeben.
Außerdem leuchten die LED´s von einem LED Stripe abhängig von der Luftfeuchtigkeit:
rot bei schlechten Werten, gelb bei erhöten Werten und grün bei guten Werten
Die LEDs leuchten in Abhängigkeit von der Helligkeit entweder mit voller Lichtstärke oder gedimmt.
Die drei Werte werden außerdem als JSON File an MQTTX gesendet und von dort aus weiter verarbeitet.
"""


# Verwendete Hardware:
"""
    ESP32
    HTU2X
    BH1750
    WS2812b
"""

# BUS-Systeme
"""
i2c:
    BH1750  (Lichtsstärke)
    HTU21   (Temperatur / Luftfeuchtigkeit)
----------
SPI:
    ST7789  (Onboard Display)
----------
neopixel:
    WS2812b
"""

```

```
# Spannungen:
"""
    BH1750 an 3,3V DC
    HTU2X an 3,3V DC
    WS2812b an 5V DC
"""

#####################################################
# Die verwendeten Bibliotheken stehen in der Readme #
#####################################################




#------------Bibliotheken aufrufen-----------------------------------------------------

import json                                 #
from main import MQTT_TOPIC, mqtt_MG        #
from time import sleep, time                #
from neopixel import NeoPixel               #
from htu2x import HTU21D                    #
from bh1750 import BH1750                   #
from machine import Pin, SoftSPI, SoftI2C   # Pin(BMP180 und TFT), SoftSPI(TFT), SoftI2C(BMP180)
import st7789py as st7789                   # TFT-Display
from fonts import vga2_16x16 as font        # Schriftart laden


i2c = SoftI2C(scl=Pin(22), sda=Pin(21))     # Pins für den i2c Bus
htu = HTU21D(22,21)                         # i2c Bus Anschluss Temperatur und Luftfeuchte
bh = BH1750(i2c)                            # i2c Bus Lichtsensor

#------------Display Initialisierung---------------------------------------------------
spi = SoftSPI(                              # Objekt spi instanzieren
        baudrate=20000000,                  # TFT Kommunikationsgeschwindigkeit
        polarity=1,                         #
        phase=0,                            #
        sck=Pin(18),                        #
        mosi=Pin(19),                       #
```

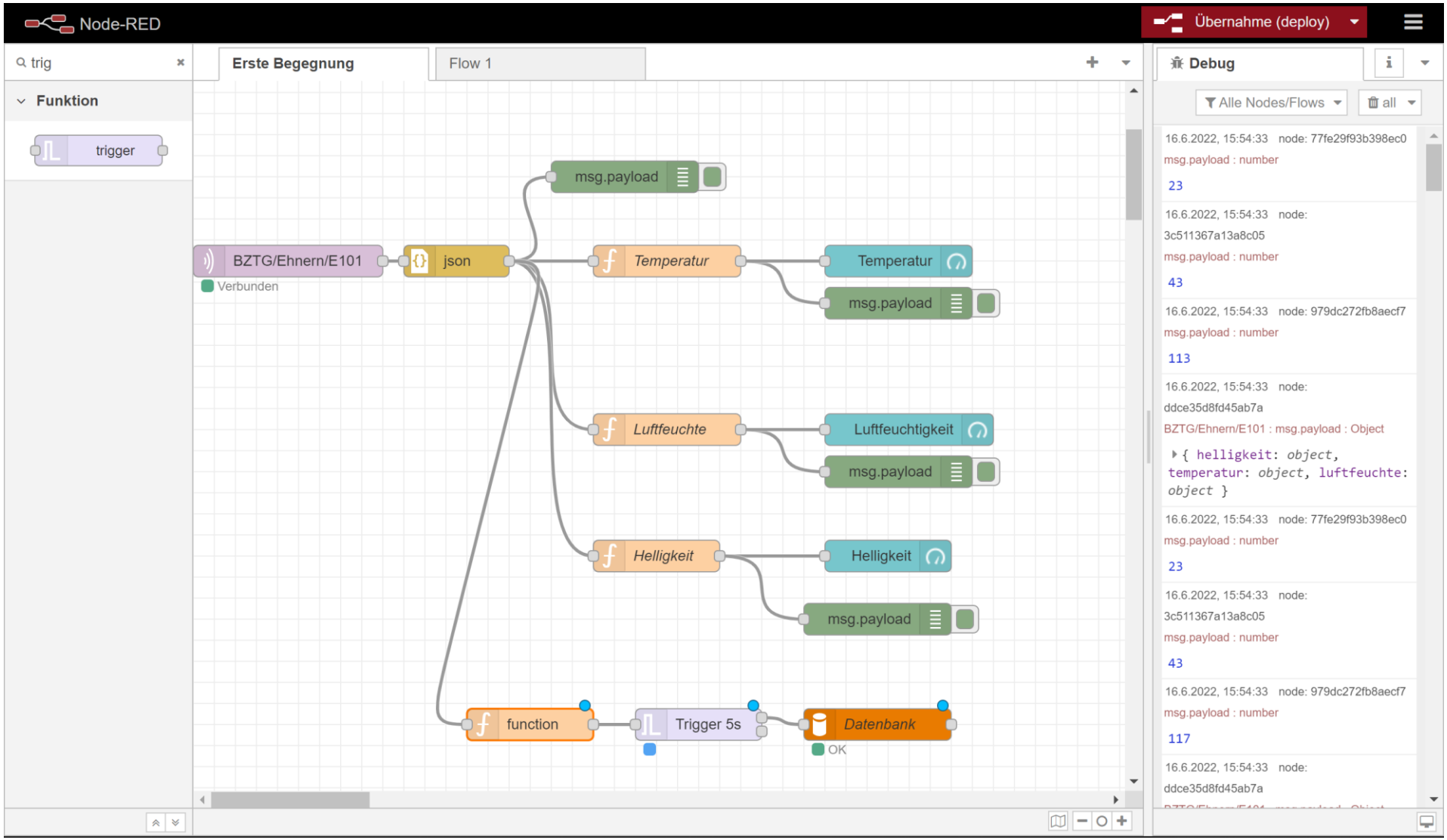
```
        miso=Pin(13))                           #
                                                #
tft = st7789.ST7789(                            # Objekt tft instanzieren
        spi,                                    # Schnittstelle
        135,                                    # Pixel x-Achse(hochkant)
        240,                                    # Pixel y-Achse
        reset=Pin(23, Pin.OUT),                 #
        cs=Pin(5, Pin.OUT),                     #
        dc=Pin(16, Pin.OUT),                    # TEST!
        backlight=Pin(4, Pin.OUT),              #
        rotation=1)                             # rotation 90°, 2 180°
#------------- Display Initialisierung Ende -----------------

#------------- Neo Pixel Initialisierung ---------------
NUM_OF_LED = 11                                 # 11 LED´s von den WS2812 LED´s verwendet
np = NeoPixel(Pin(2), NUM_OF_LED)               # LED Band an Pin 2 angeschlossen

r = 0                                           #
g = 0                                           #
b = 0                                           #

#------------| Funktionen |------------------
def clearStripe():                              #
      for i in range(NUM_OF_LED):               # Liste für die 11 LED´s
        np[i] = (0,0,0)                         #
        np.write()                              #

def neoSetzten(r, g, b):                        #
    for i in range(NUM_OF_LED):                 #
        np[i] = (r,g,b)                         #
        np.write()                              #

def convertHelligkeit(x):
    wert = (x - 0) * (254 - 1) // (3000 - 0) + 1
    if wert > 254: wert=254
    return wert


mqttSenden = 3                                  # alle 3 sekunden
```

```
119 sensorenAuswerten = 1                          # jede Sekunde
120 tftAnzeigen = 5                                # alle 5 Sekunden
121
122 zeitMqtt = time() + mqttSenden
123 zeitSensoren = time() + sensorenAuswerten
124 zeitTft = time() + tftAnzeigen                                    #
125
126 mqtt_MG.connect()                                                 #
127
128 #----------Werte senden-------------------------------
129 while True:
130     if time() >= zeitSensoren:
131         try:
132             tempa = round(htu.temperature)                        #
133             luft = round(htu.humidity)                            #
134         except:
135             print("Temp / Hum sensor kaputt")
136             temp = str(0)
137             luft = 0
138
139         try:
140             helligkeit = round(bh.luminance(BH1750.CONT_LOWRES))  #
141         except:
142             print("Helligkeit Sensor defekt")                     #
143             helligkeit = 5000                                     #
144
145         npHell = convertHelligkeit(helligkeit)                    #
146
147         if luft < 30:                                             #
148             r = 0
149             g = 1
150             b = 0
151
152         if (luft >= 30 and luft <= 50):                           #
153             r = 1
154             g = 1
155             b = 0
156
```

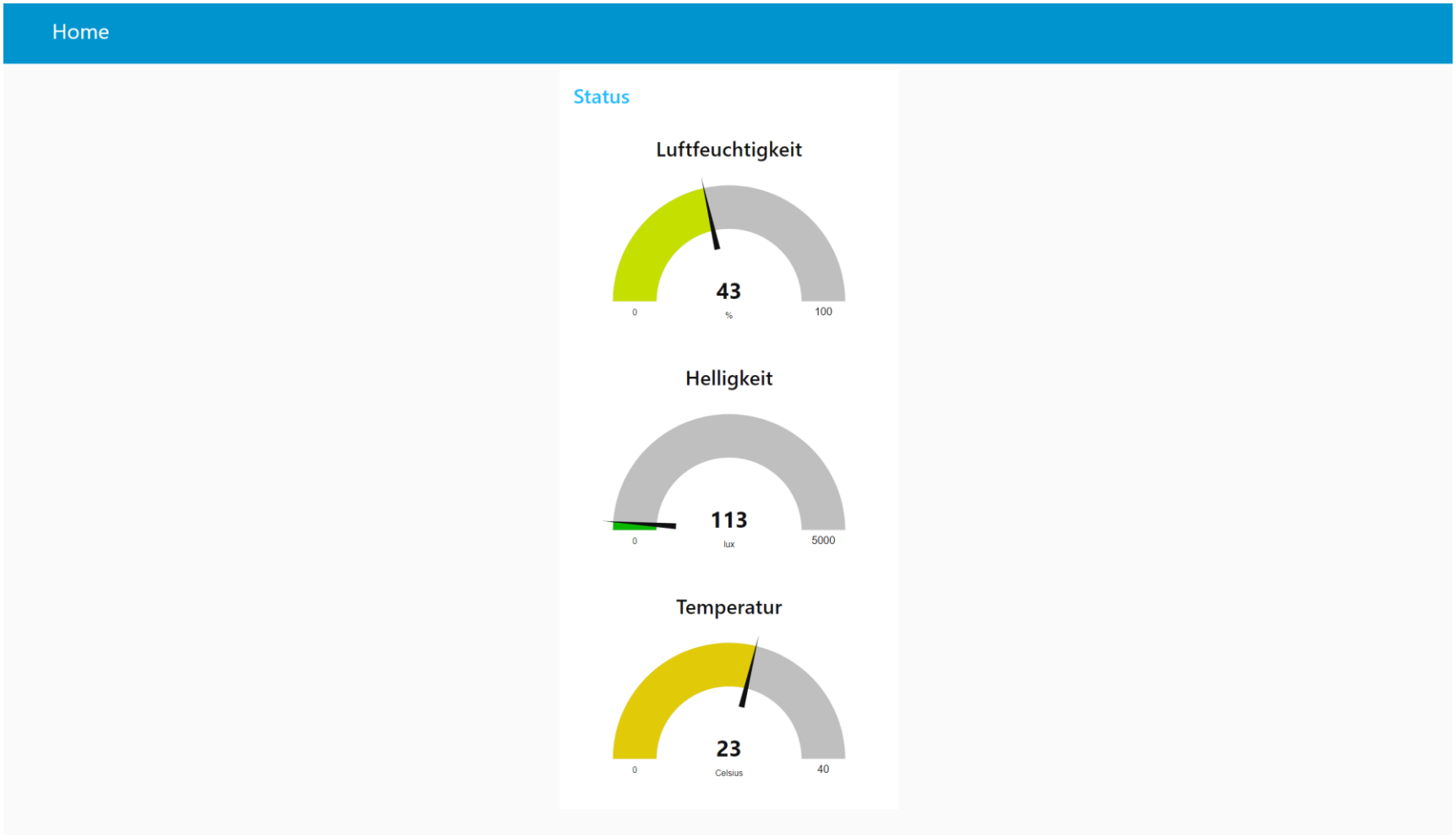
```
        if luft >= 50:                                                  #
            r = 1
            g = 0
            b = 0

        neoSetzten(r*npHell, g*npHell, b*npHell)                        #

        ZeitSensoren = time() + sensorenAuswerten                       #


    if time() >= zeitMqtt:                                              #
        datatemp = {                                                    #
            "temperatur":{                                              #
                "HTU21Dtemp": tempa                                     #
            },
            "luftfeuchte":{                                             #
                "HTU21Dluft": luft                                      #
            },
            "helligkeit":{                                              #
                "BH1750": helligkeit                                    #
            }
        }

        print("MQTT verbunden!")                                        #
        print(datatemp)                                                 #
        #mqtt_MG.connect()
        #sleep(0.5)
        mqtt_MG.publish(MQTT_TOPIC,json.dumps(datatemp))                #
        #sleep(0.5)
        #mqtt_MG.disconnect()

        zeitMqtt = time() + mqttSenden                                  #


#------------------Werte auf Display--------------------------------
    if time() >= zeitTft:                                               #
        tft.fill(st7789.BLACK)                                          #
        tft.text(font, str(tempa) + ' \xf8C', 10, 10, st7789.WHITE, st7789.BLACK)   #

        tft.text(font, str(luft) + ' %', 10, 30, st7789.WHITE, st7789.BLACK)        #

        tft.text(font, str(helligkeit) + ' lux', 10, 50, st7789.WHITE, st7789.BLACK)#

        zeitTft = time() + tftAnzeigen                                  #
```

WETTERSTATION

Node-RED

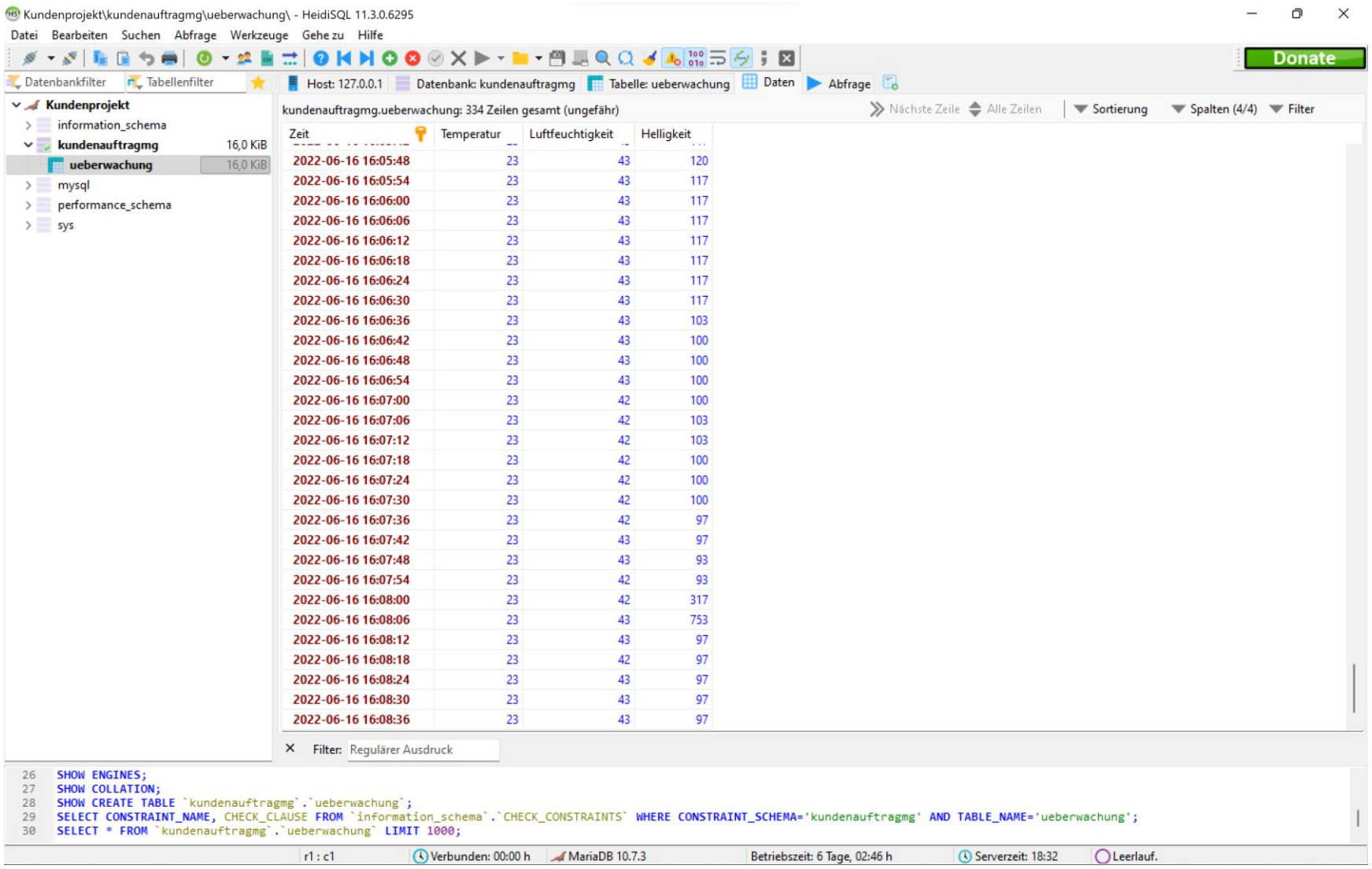


Node-RED UI

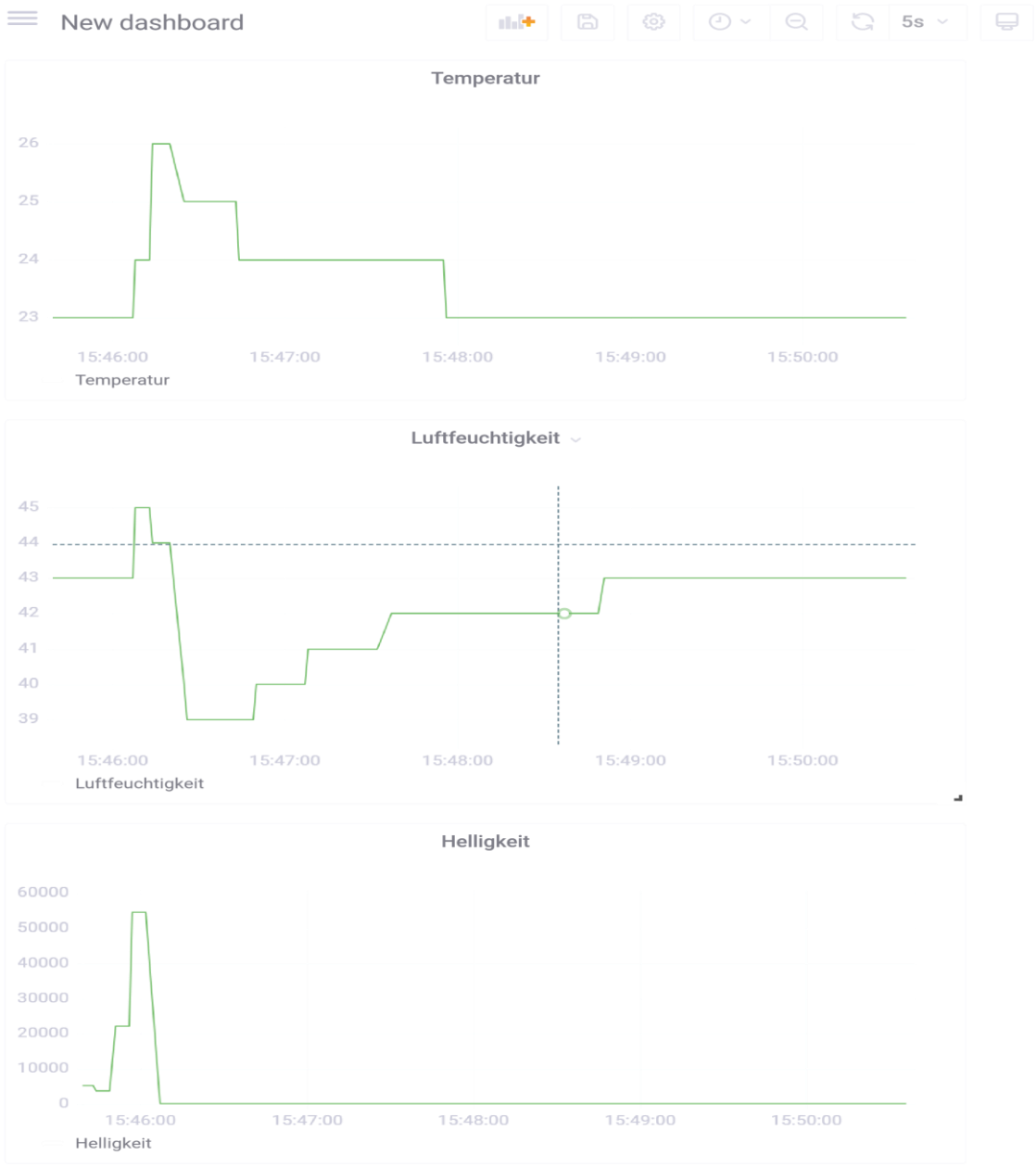

Home
Status
Luftfeuchtigkeit
43
%
0
100
Helligkeit
113
lux
0
5000
Temperatur
23
Celsius
0
40

WETTERSTATION

MySQL



Grafana


New dashboard
5s
Temperatur
26
25
24
23
15:46:00
15:47:00
15:48:00
15:49:00
15:50:00
Temperatur
Luftfeuchtigkeit
45
44
43
42
41
40
39
15:46:00
15:47:00
15:48:00
15:49:00
15:50:00
Luftfeuchtigkeit
Helligkeit
60000
50000
40000
30000
20000
10000
0
15:46:00
15:47:00
15:48:00
15:49:00
15:50:00
Helligkeit

Fazit

Da ich ja leider eine sehr lange Zeit ausgefallen bin und Prüfungen, Roboter Projekt und Präsentationen anstanden, ist mir das Projekt sehr schwergefallen. Doch dank der Hilfe von Klassenkameraden, konnte ich mein Projekt fertig stellen.

Auch konnte ich durch dieses Projekt alles was durch meine Corona Erkrankung gefehlt hat, sehr gut nachholen.

Allerdings sind die Kommentare zum Programm durch die kurze Zeit leider viel zu kurz gekommen und schlecht nachvollziehbar.

Quellen

Bildquellen (Deckblatt, Aufbau, Screenshots): Privat

HTU21: HTU21D Temperatur- und Luftfeuchtigkeitssensor I2C - Bastelgarage Elektronik Online Shop

BH1750: HALJIA gy-302 bh1750 Digital Lichtintensität Sensor Beleuchtung Detektor Modul 3 V - 5 V Kompatibel mit Arduino GY302 BH1750FVI: Amazon.de: Computer & Zubehör

ESP32: LILYGO TTGO T-Display ESP32 Development Board (16 MB) CP2104 | Elektor

WS2812b: 5m WS2812b LED Pixel Streifen 60LEDs/m IP65 (5V) | pixel-imperium.de

Bibliotheken: Siehe readme

Dokumentation, Kundenauftrag und alles weiter sind auf Github hinterlegt:

Letzter Zugriff aller Quellen: 12.06.2022 um 15 Uhr