

```

1 #include <Arduino.h>
2 #include <ESP8266WiFi.h>
3 #include <Wire.h>
4 #include <PubSubClient.h>
5 #include <Adafruit_Sensor.h>
6 #include <Adafruit_BME280.h>
7
8 #define SEALEVELPRESSURE_HPA (1013.25)
9
10 /* ##### WLAN und MQTT Zugang definieren ##### */
11
12 const char* ssid = *****;
13 const char* wifi_password = *****;
14
15 const char* mqtt_server = *****;
16 const char* mqtt_username = *****;
17 const char* mqtt_password = *****;
18
19 /* ##### */
20
21
22 /* ##### MQTT Topics bestimmen ##### */
23
24 const char* Voltage_topic = "home/Voltage_Test/Spannung";
25 const char* Temperatur_topic = "home/Voltage_Test/Temperatur";
26 const char* Feuchtigkeit_topic = "home/Voltage_Test/Feuchtigkeit";
27 const char* Ladung_topic = "home/Voltage_Test/Ladung";
28 const char* clientID = "client_Voltage_Test";
29
30 /* ##### */
31
32 WiFiClient wifiClient;
33 Adafruit_BME280 bme;
34 PubSubClient client(mqtt_server, 1883, wifiClient);
35
36 //##### Variablen
37
38 unsigned int raw = 0;
39 float volt = 0.0; //Kennzahl Volt
40 float ladung = 0; //Kennzahl Ladung in %
41 float adcwert = 0.0, radcwert = 0.0;
42 float calibration = 0.15; //A0 Kalibration
43 const int sleepTimeS = 60; //Schlafzeit in Sek.
44 double faktor = 5.4054054054054054054054054054054; //Faktor basierend auf Widerstandswerten
45 float h, t; //Variablen für Temperatur und Feuchtigkeit
46
47 //Programm Anfang
48
49 //##### Verbindungsaufbau und MQTT Verbindung
50 void connect_MQTT(){
51   Serial.print("Connecting to ");
52   Serial.println(ssid);
53
54   WiFi.begin(ssid, wifi_password);
55
56   while (WiFi.status() != WL_CONNECTED) {
57     delay(500);
58     Serial.print(".");
59   }
60
61   Serial.println("WiFi Verbunden");
62   Serial.print("IP adresse: ");
63   Serial.println(WiFi.localIP());
64
65   if (client.connect(clientID, mqtt_username, mqtt_password)) {
66     Serial.println("Verbunden mit MQTT Broker!");
67   }
68   else {
69     Serial.println("Verbindung mit MQTT Broker fehlgeschlagen...");
70   }
71 }
72
73 //##### AnalogPin auslesen und umrechnen
74 void handleIO() {
75
76
77   volt = analogRead(A0) * faktor / 1023;
78
79   volt = volt + calibration;
80
81   ladung = (volt-3)/1.2 * 100.00;
82
83 }
84
85
86 //##### BME280 Daten auslesen
87
88 void getWeather(){
89
90   Serial.println(F("BME test"));
91   bool status;
92
93   status = bme.begin(0x76);
94   if (!status) {
95     Serial.println("Kann keinen BME280 finden....");
96   }
97
98   t = bme.readTemperature();
99   h = bme.readHumidity();
100
101
102
103
104
105 }
106
107 void setup() {

```

```

108     Serial.begin(115200);
109
110
111 }
112
113
114
115 void loop() {
116     connect_MQTT();
117     Serial.setTimeout(2000);
118     handleIO();
119     getWeather();
120
121     Serial.print("Spannung ");
122     Serial.print(volt);
123     Serial.println(" V");
124     Serial.print("\n");
125     Serial.print("Temperatur ");
126     Serial.print(t);
127     Serial.println(" *C");
128     Serial.print("Feuchtigkeit ");
129     Serial.print(h);
130     Serial.println(" %");
131
132
133
134
135
136     //Daten in String umwandeln für MQTT
137     String vs="Volt: "+String((float)volt)+" V ";
138     String ts="Temp: "+String((float)t)+" C ";
139     String hs="Hum: "+String((float)h)+" % ";
140     String ls="Ladung: "+String((float)ladung)+" % ";
141
142     //Daten an MQTT Broker senden
143     //Spannung
144     if(client.publish(Voltage_topic, String(volt).c_str())){
145         Serial.println("Spannung gesendet! \n");
146     }
147     //Neuersuch bei Fehlerhafter Verbindung (client.publish gibt ein boolean zurück)
148     else {
149         Serial.println("Senden fehlgeschlagen. Verbinde neu und sende erneut.....");
150         client.connect(clientID, mqtt_username, mqtt_password);
151         delay(15);
152         client.publish(Voltage_topic, String(volt).c_str());
153     }
154     //Temperatur
155     if(client.publish(Temperatur_topic, String(t).c_str())){
156         Serial.println("Temperatur gesendet! \n");
157     }
158     //Neuersuch bei Fehlerhafter Verbindung (client.publish gibt ein boolean zurück)
159     else {
160         Serial.println("Senden fehlgeschlagen. Verbinde neu und sende erneut.....");
161         client.connect(clientID, mqtt_username, mqtt_password);
162         delay(15);
163         client.publish(Temperatur_topic, String(t).c_str());
164     }
165     //Feuchtigkeit
166     if(client.publish(Feuchtigkeit_topic, String(h).c_str())){
167         Serial.println("Feuchtigkeit gesendet! \n");
168     }
169     //Neuersuch bei Fehlerhafter Verbindung (client.publish gibt ein boolean zurück)
170     else {
171         Serial.println("Senden fehlgeschlagen. Verbinde neu und sende erneut.....");
172         client.connect(clientID, mqtt_username, mqtt_password);
173         delay(15);
174         client.publish(Feuchtigkeit_topic, String(h).c_str());
175     }
176     //Ladung
177     if(client.publish(Ladung_topic, String(ladung).c_str())){
178         Serial.println("Ladung gesendet! \n");
179     }
180     //Neuersuch bei Fehlerhafter Verbindung (client.publish gibt ein boolean zurück)
181     else {
182         Serial.println("Senden fehlgeschlagen. Verbinde neu und sende erneut.....");
183         client.connect(clientID, mqtt_username, mqtt_password);
184         delay(15);
185         client.publish(Ladung_topic, String(ladung).c_str());
186     }
187
188
189     client.disconnect(); //Verbindung vom MQTT broker trennen
190     ESP.deepSleep(9e8);
191     //delay(1000*60); //Sende jede Minute neue Daten
192 }

```