Designing a weather station using a BME280 sensor and ESP8266 NodeMCU-E12 Arduino board with connection to Thingspeak™ cloud

Again, I want to express my gratitude to the global community for sharing many useful information. It was relatively **easy as a beginner to connect a sensor BME 280 to the Arduino ESP8266 NodeMCU-E12** a**nd send data to the cloud**.

Important - please read: Technical advice and document disclaimer

The following examples, recommendations, software codes, wirings and installation hints or any descriptions in this report or attached pdf shall be used for general information only. Internet links provided to other sources of information are subject to the use own considered pre-cautions and risks. The author believes that the content is based on sound and established principles, they shall not be deemed as appropriate for any specific application or project without subsequent independent analysis, evaluation, verification and assessment of appropriateness for such an application by the building person using this for own projects, since the author cannot be considered to be nor to become a specialist. Based on the preceding, the author hereby declines any liability whatsoever for losses and/or damages of whatever kind (and sustained by whomever) that might result from this document.

Find in this document further information and links about:

1. Pre-requisites: Installation of Arduino IDE Software and library for ESP8266 NodeMCU-E12
2. Sample sketches for the weather station
3. Calibration hints for the sensor BME280
4. Basic information about Thingspeak
5. How to connect to Thingspeak?
6. How to visualize the data?
7. About the measurement unit
8. Placing the sensor unit in a weather hut

1. Pre-requisites: Installation of Arduino IDE Software and library for ESP8266 NodeMCU-E12

If not yet installed on your PC, please download the software editor **Arduino IDE** here: https://www.arduino.cc/en/Guide/HomePage . Then use one of the provided pdf document (only sample sketches) and copy the complete text into the Arduino editor.

Please ensure all libraries needed and listed in the header of the program (**example: #include <Wire.h>**) are installed (for installation information of the ESP8266 NodeMCU-E12 search the web. An example can be found here: https://www.instructables.com/id/Quick-Start-to-Nodemcu-ESP8266-on-Arduino-IDE/).

2. Sample sketches for the weather station:

Please find attached two basic versions of a running program, so called “sketches” (in this section of the website):

2.1 One sketch for **simply running connected to the PC via USB port** and get the data for temperature, humidity and barometrical pressure displayed via the Arduino software tool “Monitor”. Make sure the Baud rate matches from Serial.begin inside the sketch with the status line of the monitor.

2.2 The second sketch provides is an already **advanced version with WLAN-connection with sending the data to the cloud “Thingspeak”**. This is not an need to be used since there are other providers and “clouds” to be used as well. Find further needed steps below.

In addition, it might be helpful to insert the Arduino OTA (Over the Air) WLAN upload capability of software sketch updates to the remote situated ESP8266 unit. Additional software must be installed on the PC. Find more information as one example source here: http://arduino.esp8266.com/Arduino/versions/2.0.0/doc/ota_updates/ota_updates.html

3. Calibration hints for the sensor BME280:

Inside the sketches you will find variables with “calibration” values, which you may use to set slight adjustments. Setting them to “1.0” gives the original values of the BME280 connected.

I found it necessary to define such corrective multipliers, since in comparison with a reliable measurement devise, e.g., the **Peaktech 5185**, https://www.peaktech.de/productdetail/kategorie/datenlogger-temperatur/produkt/peaktech-5185.html deviations eventually can be corrected for the BME280 data.

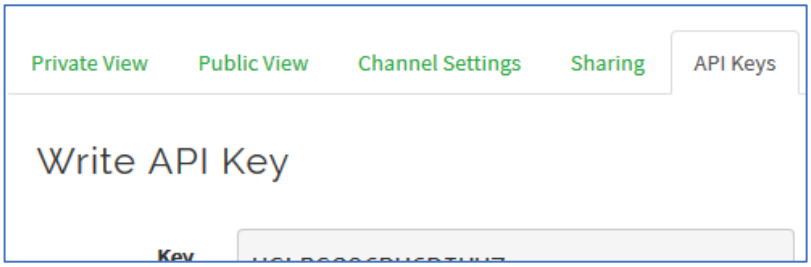
They vary among produced BME280 devises, but in general provide reliable data. The barometrical pressure was calibrated based on the values given by the nearest official weather station found in the web and adjusted on the altitude difference of the official station and the altitude of my own sensor. Simplified one can use the barometric pressure difference of “1hPa per each 8m” (only valid up to approximately 5000 m). Just calculate the corrective multiplier related to the altitude of your station.

4. Basic information about Thingspeak:

The data cloud is free in use up to certain limits of data volume and channels, but submitting weather data every five minutes this is more than enough. https://thingspeak.com/ **After creating an account and signing in, you can create a new channel (meaning to define the connection to your sensor data).** For the three values of temperature, humidity and barometric pressure you can define the three “fields” and give them your own description with a meaningful short name, e.g., “TempGarden”.

5. How to connect to Thingspeak?

Just follow **these steps to create your new channel** and find under API Keys the Key to write data. https://de.mathworks.com/help/thingspeak/collect-data-in-a-new-channel.html. The channel and write API key information is provided on the page of your selected channel.



On completion you need to insert the given **write API-key of the channel** into the Arduino sketch with WLAN and Thingspeak, e.g, **line 17**. "Writing" means sending the data to the cloud.

In addition, you need to **modify the lines 18 & 19 to get access to your own Router by inserting the your valid SSID and password**.

```
16 // replace with your channel's thingspeak API key,
17 String apiKey = "YOUR-API-WRITE-KEY";      // den Key für write API aus Thingspeak eintragen
18 const char* ssid = "YOUR SSID";       // Router
19 const char* password = "YOUR PASSWORD";    // Passwort
```

After saving, compiling and uploading to the Arduino sketch, your sensor data are transmitted to the cloud. For testing set the cycle time to at least 15000 ms and later in normal run mode e.g., to 300000 ms, which corresponds to 5 minutes. Please use the Arduino Monitor tool to check the program results being displayed.

6. How to visualize the data?

This can be achieved for example in three different ways.

6.1 The fast way is to use the **existing Thingspeak account** by selecting to your created channel and add pre-defined widgets and visualization of charts gives fast results. Select your channel and select the button "+widget or +visualizations": https://de.mathworks.com/help/thingspeak/channel_display_widgets.html

Follow the process and then customize your settings of the charts for style, title, data number displayed, min-max figures. Important: the field number you "tick" and define with a freely selectable name must correspond to the Arduino sketch "field" that you send, e.g., TEMP_ROUND in Arduino can be named "TempGarden" in Thingspeak.

Reference them from the Arduino sketch:

```
94              String postStr = apiKey;
95
96                  postStr +="&field1=";
97                  //postStr += String(bme.readTemperature()
98                  TEMP_ROUND = round ((TEMP*TempCor)*10)/10
99                  // postStr += String(TEMP*0.905963); Varia
100                 postStr += String(TEMP_ROUND);
101
102
103                 postStr +="&field2=";
```

6.2 The **second option is a really cool App** for mobile devises is: ThingView – just **check it out on the App store in the internet** for either use with iPhone or Android! You simply connect the App with your account and with "+" add a new channel (insert the Channel ID (find it it in your Thingspeak Channel information) and the Server URL: https://thingspeak.com) to receive all the "field" data in charts or widgets.

6.3 If you are using an own website, you can use JAVA Script designed widgets from https://www.highcharts.com/ . Using for example JSON commands to "read" the data from your cloud storage Thingspeak and visualize them.

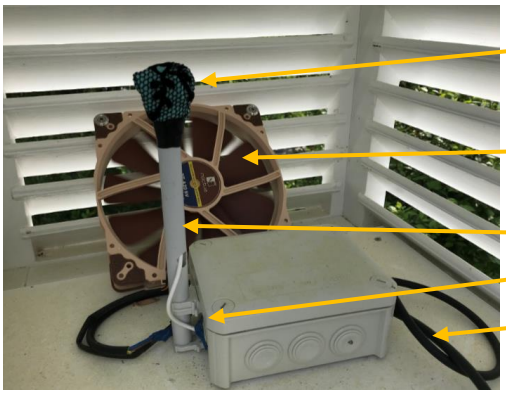
You need to insert commands with $getJSON… and need to provide for access the channel-ID number, the wanted field number and the "READ API Key" (which is all given in your created Thingspeak channel) in order to download serie data from the cloud.

That is a bit more effort of course and many sources are found there or given by the community. https://www.highcharts.com/forum/viewtopic.php?t=40384

7. About the measurement unit:

7.1 Where to put the electronics if used outdoor? The ESP8266 module needs to be placed in a water protected installation box, type rating IP67. The cable should be of outdoor capability (rubber coated not PVC!) and is connected through the cable inlets of the box from to the related pins of BME280. In this solution, the cable runs through an electric installation pipe, clamped outside the box and with a length of approximately 15 cm height the sensor is located then above the box. The cables are soldered to the provided pin-set female version of the BME280. **Remark: The sensor needs to be situated 2m above ground level.**

The design looks as follows:

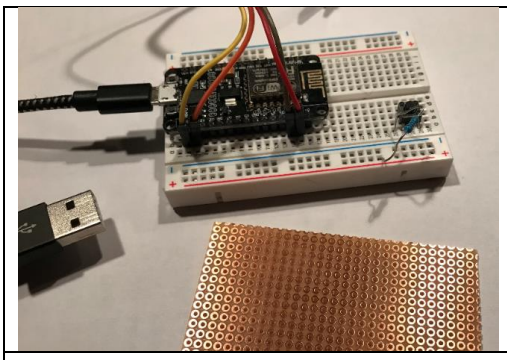


- BME280 covered with non-cotton material (e.g., nylon mesh)
- 5V fan for ventilation of the weather hut, NOCTUA
- Pipe with clamps, glued to the IP67 rated installation box
- Outdoor, rubber coated cable and power inlet for 5V DC

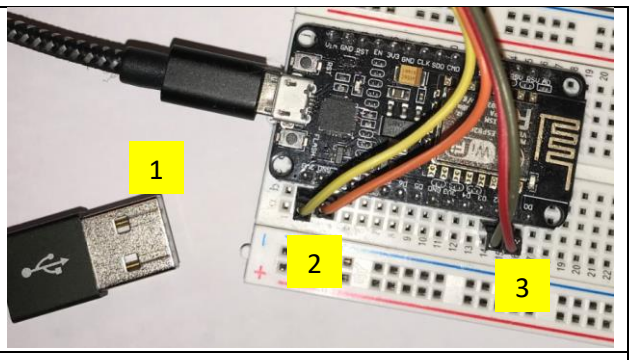
Warning: **Working with electronic or electrical equipment can be dangerous and harmful. Only licensed electrical workers can legally perform electrical work. For safety reasons, the power cable running to the weather hut must be outdoor capable and -rated. As well the e.g. 230 AC to 5V DC power conversion must be done in a water protected indoor location** (e.g., a USB charger with certified quality and electrical low noise, 1A rating, otherwise the ESP8266 and fan may not work correctly). **Use the expertise of a certified specialist to avoid mistakes, hazards, damages or even electrical shocks!**

The installed fan for ventilation inside the weather hut is as well a 5V DC fan and is switched via a low-voltage relay as mentioned below.

Building the application:



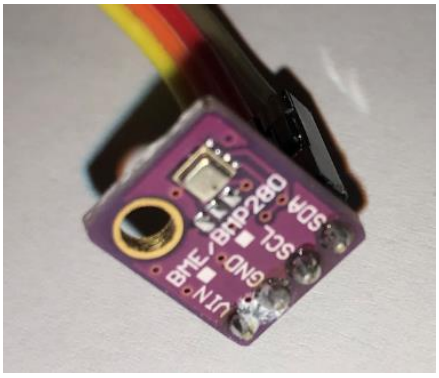
The ESP8266 unit after tested on the board, is soldered for the final project to a perfboard (Lochrasterplatine). Please solder for easy maintenance plug connectors first to the prefboard and then simply plug-in the ESP8266 then.



1: USB connection to PC

2: Power supply to BME280:
left 3,3V (yellow) → BME280: VIN
right GND (orange) → BME280: GND

3: Data connection to BME:
left D2, (brown) → BME280: SDA
right D1, (red) → BME280: SCL

	 BME280 connection side: Note a plug connector 4-pin is provided normally with each sensor. Use special cables with plug to connect the sensor for easy exchange if needed later.

8. Placing the sensor unit in a weather hut

If used outdoor, a weather protection is required, meaning e.g., a so called “English Weather hut”. A self-made version can be done by using wooden window shutters with wide spacing as ventilated “box” and painted out- and inside all white. The size in my case is 65 cm in height and is 54 cm in width. Find more official information here https://de.wikipedia.org/wiki/Thermometerh%C3%BCtte .

My findings: The electronic and program incl. Thingspeak connection was ready in 3 days - the design of the weather hut incl. installation in the garden took me 3 months. 😉

More findings including the need for using a ventilation fan are described here: **https://koch-manfred.de/Wetterstationen.** The weather hut must be ventilated to achieve reliable data, especially at low wind conditions, when radiation of the sun is heating the material up. In the meantime, a modified, not provided Arduino sketch is in use to trigger an output and switch a low voltage relay for turning the 5V DC-fan on- and off based on daylight and temperature conditions.

Good luck and have fun!