# BME280-Breakout von watterott.com



H39@email.de

| BME280-Breakout von watterott.com | |
|---|---|
| www.watterott.com | BME280-Breakout (Luftfeuchtigkeits-, Druck & Temperatursensor) |
| | Der BME280 ist einer der neuesten Luftfeuchtigkeits-, Druck- und Temperatursensoren von Bosch mit einem digitalen I2C und einem SPI Interface.<br>Auf dem Breakout befinden sich ein Spannungsregler und ein Pegelwandler für die I2C/SPI Schnittstelle, daher kann der Sensor von 3V - 5,5V betrieben werden. |
| Features |  <br>• Humidity sensor<br>• Pressure sensor<br>Pressure range 300 … 1100 hPa<br>• Temperatur Sensor<br>Operating range Operational -40°C - +85°C |
| Weitere Infos | github.com/watterott/BME280-Breakout |
| Quelle: | https://www.watterott.com/de/BME280-Breakout-Luftfeuchtigkeits-Druck-Tempertursensor |

| Libraries installieren | |
|---|---|
| | Bevor der BME280 mit der Arduino-IDE programmiert werden kann, müssen 2 Libraries heruntergeladen, entpackt und installiert werden. |
| Arduino Library and Examples | https://github.com/adafruit/Adafruit_BME280_Library<br>https://github.com/adafruit/Adafruit_Sensor |
| Achtung | Das installieren der „Adafruit_BME280_Library“ reicht nicht aus. Zusätzlich muss noch die „Adafruit_Sensor“ Library installiert werden. |
| Empfehlung manuelle Installation | Finde den Speicherort der Arduino-Installation heraus.<br>Um Probleme mit Schreibrechten zu vermeiden, ist es ratsam die Arduino-IDE unter „C:\Users\Public“ zu speichern. Hier sind stets Schreibrechte vorhanden.<br>Der Pfad zum Library-Ordner lautet dann:<br>*„C:\Users\Public\Programme\Arduino-1.8.5\libraries“* |
| Installation | Beide Libraries entpacken und in den „libraries-Ordner“ der Arduino-IDE kopieren. |
| | Die Arduino-DIE neu starten. |
| Link „Documentation on learn.watterott.com“: | http://learn.watterott.com/sensors/bme280/ |

## Arduino UNO & Sketch „bme280test.ino“

| | |
|---|---|
| Arduino-IDE starten<br>Sketch öffnen | Datei > Beispiele > „Adafruit BME280 Library“ > „**bme280test.ino**“ öffnen: |
| Zeilennummern aktivieren | Datei > Voreinstellungen > „Häkchen in Zeilennummern anzeigen“ |
| Betriebsarten | • Hardware SPI (Serial Peripheral Interface) mit SCK, MISO, MOSI, SS<br>• Software SPI (Serial Peripheral Interface) mit SCK, MISO, MOSI, SS (wenn man DPINS neu zuordnen will)<br>• I2C |

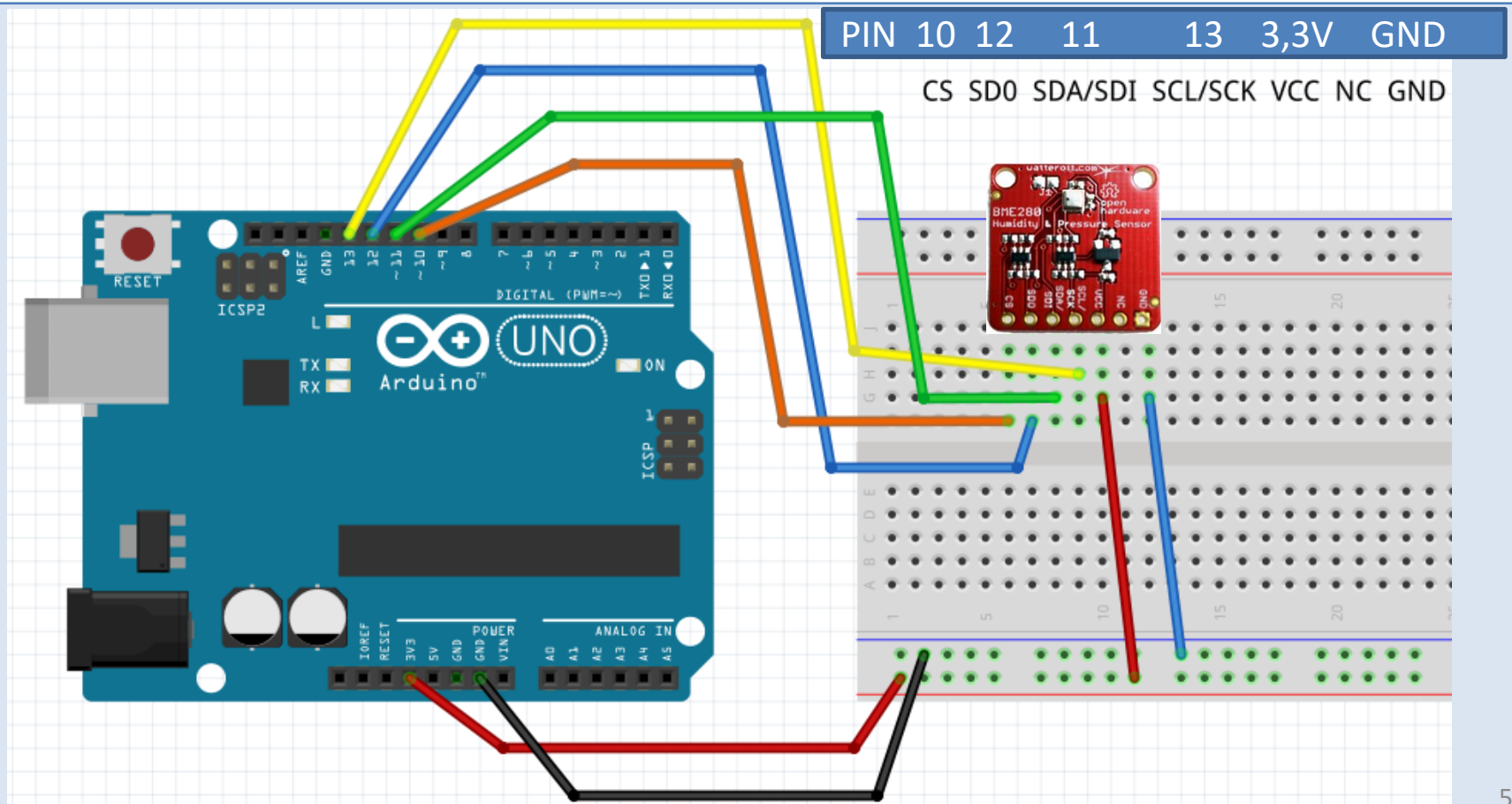
| Für Hardware-SPI sind festgelegt: | Board | MOSI | MISO | SCK | SS |
|---|---|---|---|---|---|
| | UNO | 11 | 12 | 13 | 10 |

In der „bme280test.ino“ Die Zeilen 23 bis 26 mit zusätzlichen Kommentaren:

```
22 //SPI-wiring            //  BME280                      Arduino UNO
23 #define BME_SCK 13       //  SCL/SCK   (Clock)           DPIN 13 or SCK   Serial-Clock
24 #define BME_MISO 12      //  SD0       (Serial Data Out) DPIN 12 or MISO  Master-In Slave-Out
25 #define BME_MOSI 11      //  SDA/SDI   (Serial Data In)  DPIN 11 or MOSI  Master-Out Slave-In
26 #define BME_CS 10        //  CS        (Chip-Select)     DPIN 10 or SS    Slave-Select
```

Quelle https://www.arduino.cc/en/Reference/SPI

Arduino UNO & Hardware-SPI-Betriebsart; Schaltbild

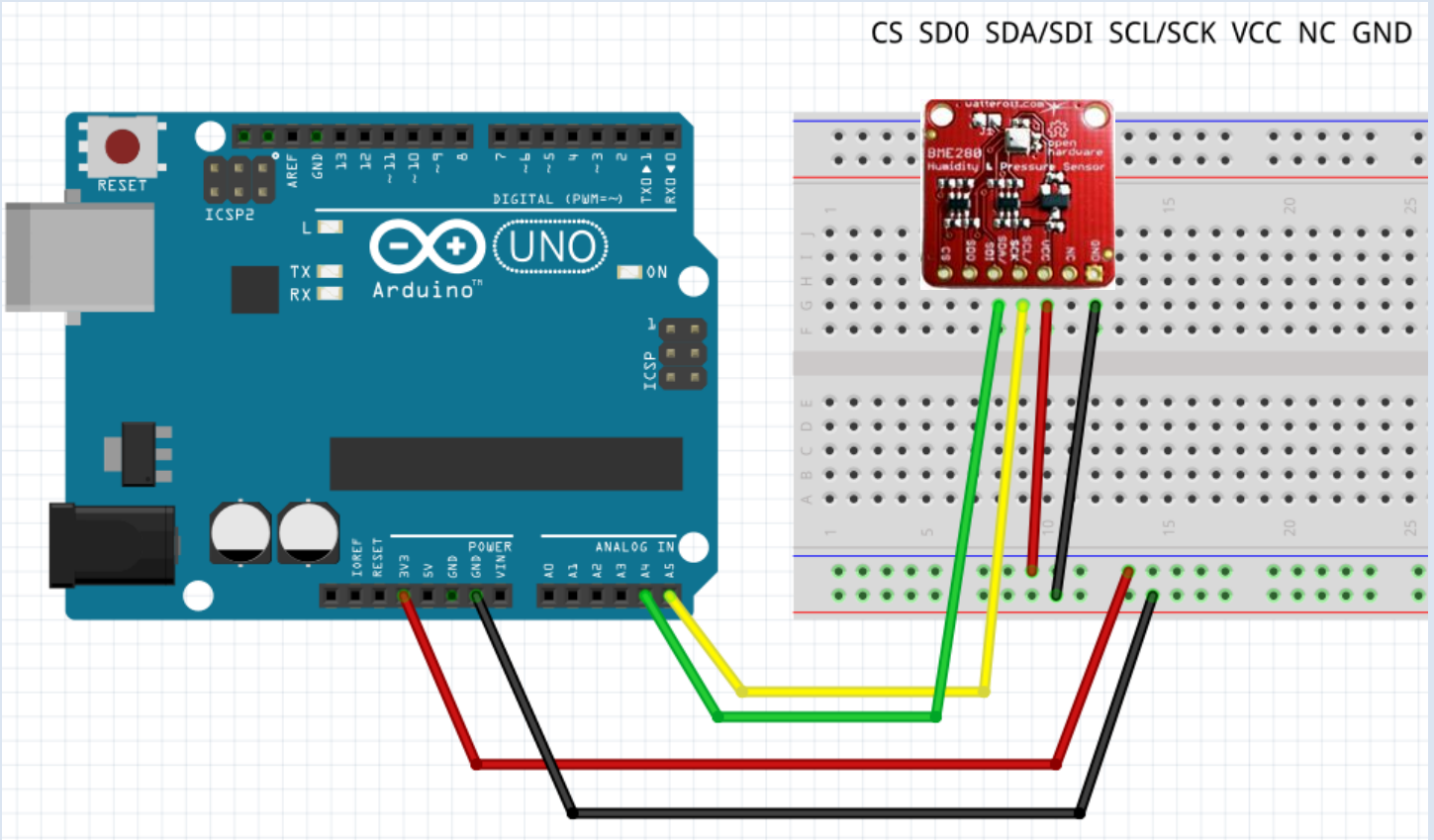


| Arduino UNO & „bme280test.ino“ in der Hardware-SPI-Betriebsart | | |
|---|---|---|
| | Die Betriebsart „Hardware-SPI“ mit dem Arduino-UNO testen. | |
| Wertvolle Tipps in | https://learn.adafruit.com/adafruit-bme280-humidity-barometric-pressure-temperature-sensor-breakout/arduino-test | |
| I2C in Zeile 30 deaktivieren (Kommentar setzen) | 30 *//Adafruit_BME280 bme; // I2C* | |
| Hardware-SPI in Zeile 31 aktivieren (Kommentar entfernen) | 31 *Adafruit_BME280 bme(BME_CS); // hardware SPI* | |
| Besonderheit | Hier müssen die auf dem jeweiligem Board für SCK, MISO, MOSI und SS vorgesehenen DPINS verdrahtet werden! „CS“ kann ein anderer DPIN sein. | |
| Hardware-SPI Anschlüsse | BME2801 | Arduino UNO |
| | VCC<br>GND<br>SCL/SCK<br>SD0<br>SDA/SDI<br>CS | 5V oder 3,3 V<br>GND<br>DPIN 13<br>DPIN 12<br>DPIN 11<br>DPIN 10 |

| Arduino UNO & „bme280test.ino“ in der Software-SPI-Betriebsart | | |
|---|---|---|
| | Die Betriebsart „Software-SPI“ mit dem Arduino-UNO testen. | |
| I2C in Zeile 30 deaktivieren mit Kommentaren | 30 *//Adafruit_BME280 bme; // I2C* | |
| Software SPI in Zeile 32 aktivieren (Kommentar entfernen) | 32 *Adafruit_BME280 bme(BME_CS, BME_MOSI, BME_MISO, BME_SCK); // software SPI* | |
| Besonderheit | Da die DPINS in Zeile 32 definiert werden (Objekt „bme), sind auch andere DPINS möglich! | |
| Software-SPI Anschlüsse | BME2801 | Arduino |
| | VCC<br>GND<br>SCL/SCK<br>SD0<br>SDA/SDI<br>CS | 5V oder 3,3 V<br>GND<br>DPIN 11 // oder Alternative<br>DPIN 10 // oder Alternative<br>DPIN 9 // oder Alternative<br>DPIN 8 // oder Alternative |

| I2C-Betriebsart | | | |
|---|---|---|---|
| Wertvolle Tipps in | https://learn.sparkfun.com/tutorials/i2c | | |
| | https://tronixstuff.com/2010/10/20/tutorial-arduino-and-the-i2c-bus/ | | |
| I2C | Die Idee ist, dass über 2 Leitungen, SDA und SCL genannt, mehrere Geräte kommunizieren können. Daher auch I2C-Bus. | | |
| | Dazu wird z.B. ein Temperatursensor in einem „Breakout Board“ integriert, das I2C bereit stellt, erkennbar an den Kontakten SDA und SCL. | | |
| Für I2C sind festgelegt: | Board | SDA (data) | SCL (clock) |
| | Arduino UNO | A4 | A5 |
| | Arduino Mega | DPIN 20 | DPIN 21) nicht getestet! |
| | ESP-12E nodeMCU | D2 | D1 |
| | ** ESP-01 | GPIO0 | GPIO2 |
| ** Anderer DPIN mit | Wire.begin( SDA, SCL) im Setup() einfügen und definieren.<br>Wire.begin( 0, 2); | | |
| Quelle | https://www.arduino.cc/en/Tutorial/MasterReader | | |

**Arduino UNO** in der I2C-Betriebsart; Schaltbild




A4 an SDA/SDI und A5 an SCL/SCK

**Arduino UNO** & „bme280test.ino“ in der I2C-Betriebsart

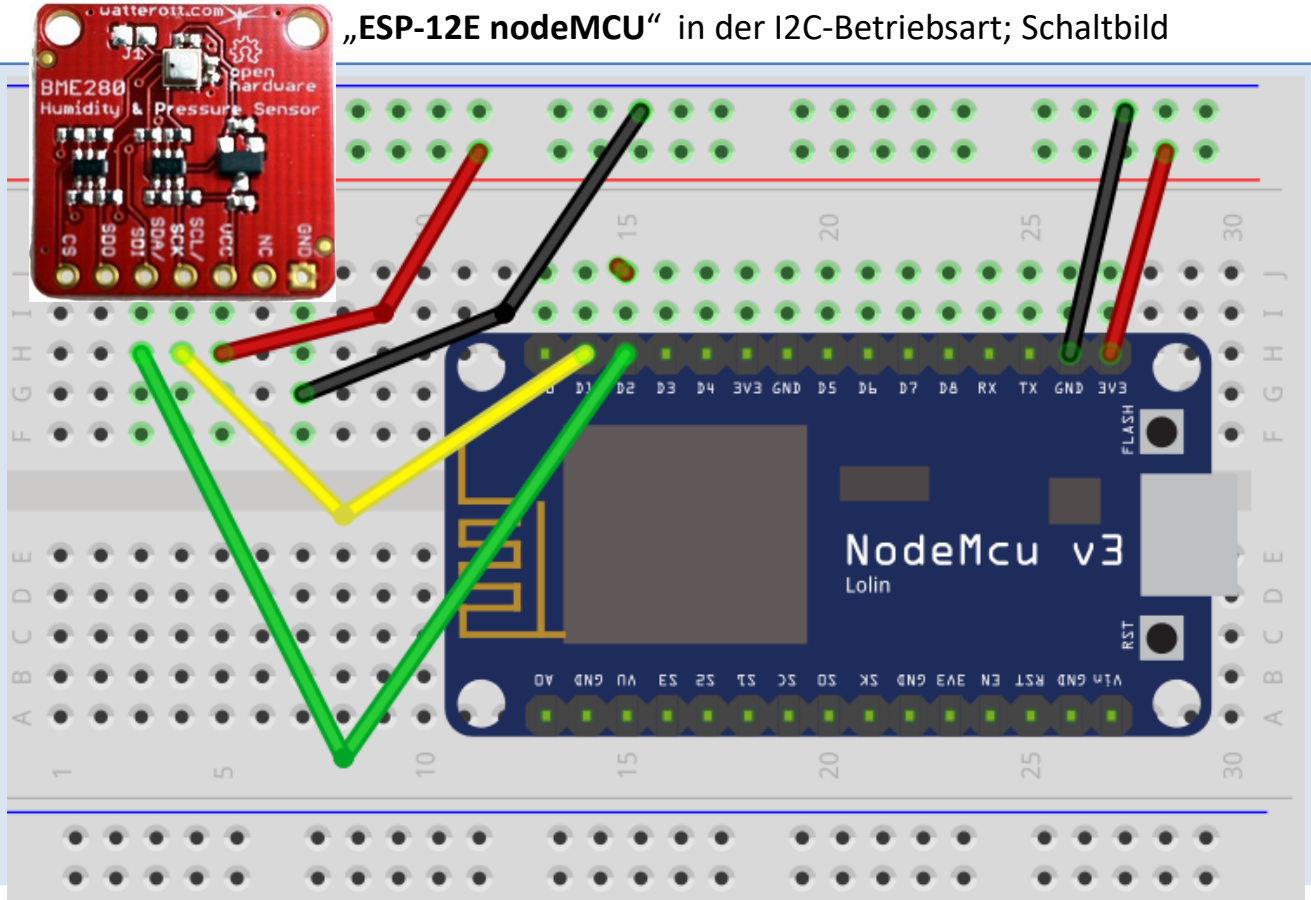
| | |
|---|---|
| Wertvolle Tipps in | https://learn.adafruit.com/adafruit-bme280-humidity-barometric-pressure-temperature-sensor-breakout/arduino-test |
| Die Zeilen 23 bis 26 mit Kommentaren | Da im I2C-Modus die SPI-DPIN-Zuordnung nicht benötigt wird, kann sie auskommentiert werden. |

```
22 //SPI-wiring              //  BME280                        Arduino UNO
23 //#define BME_SCK 13      //  SCL/SCK   (Clock)             DPIN 13 or SCK   Serial-Clock
24 //#define BME_MISO 12     //  SD0       (Serial Data Out)   DPIN 12 or MISO  Master-In Slave-Out
25 //#define BME_MOSI 11     //  SDA/SDI   (Serial Data In)    DPIN 11 or MOSI  Master-Out Slave-In
26 //#define BME_CS 10       //  CS        (Chip-Select)       DPIN 10 or SS    Slave-Select
```

| Verkabelung: | Board | SDA (data) | SCL (clock) |
|---|---|---|---|
| | Arduino UNO | A4 | A5 |

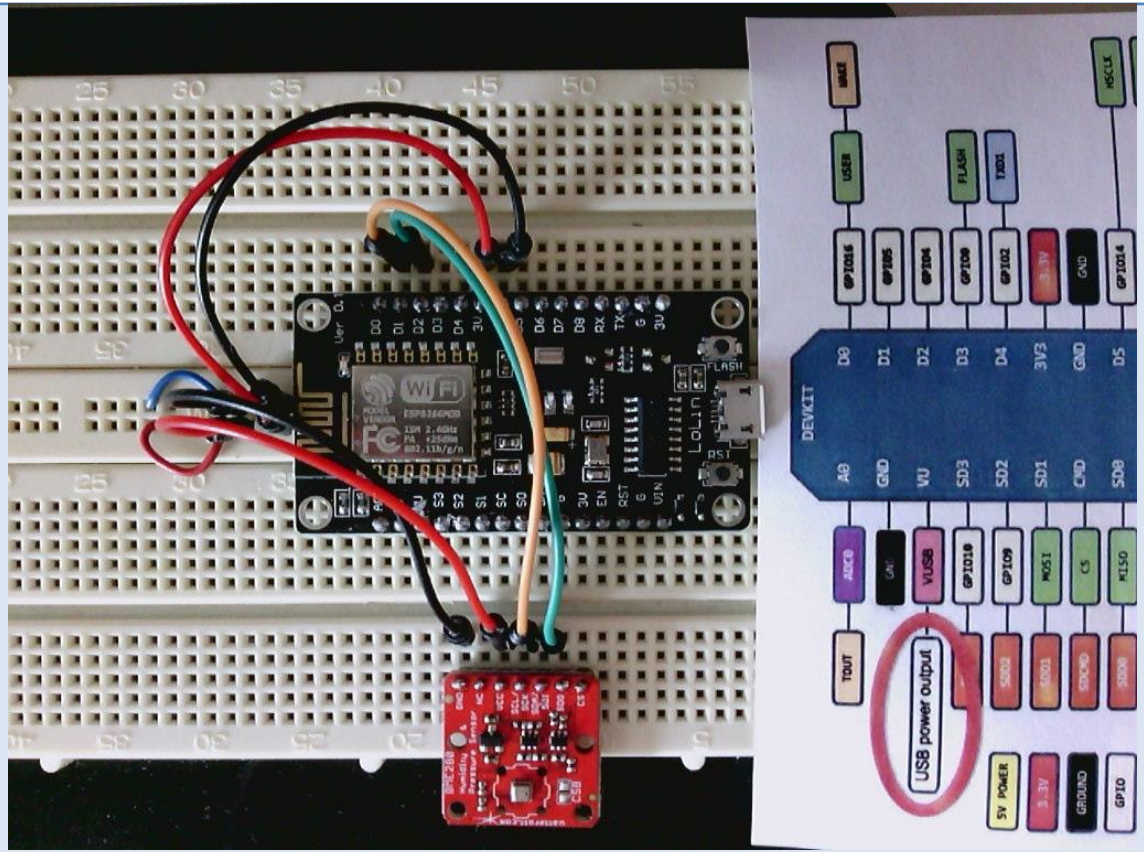
| | |
|---|---|
| I2C in Zeile 30 aktivieren (kein Kommentar) | 30 *Adafruit_BME280 bme; // I2C* |
| | Besonderheit: Hier müssen die auf dem jeweiligem Board für SCL und SDA vorgesehenen PINS verdrahtet werden!<br>Beim Arduino-UNO sind das die analogen PINS APIN 4 und APIN 5. |

„**ESP-12E nodeMCU**“ in der I2C-Betriebsart; Schaltbild




DPIN2 an SDA/SDI und DPIN1 an SCL/SCK

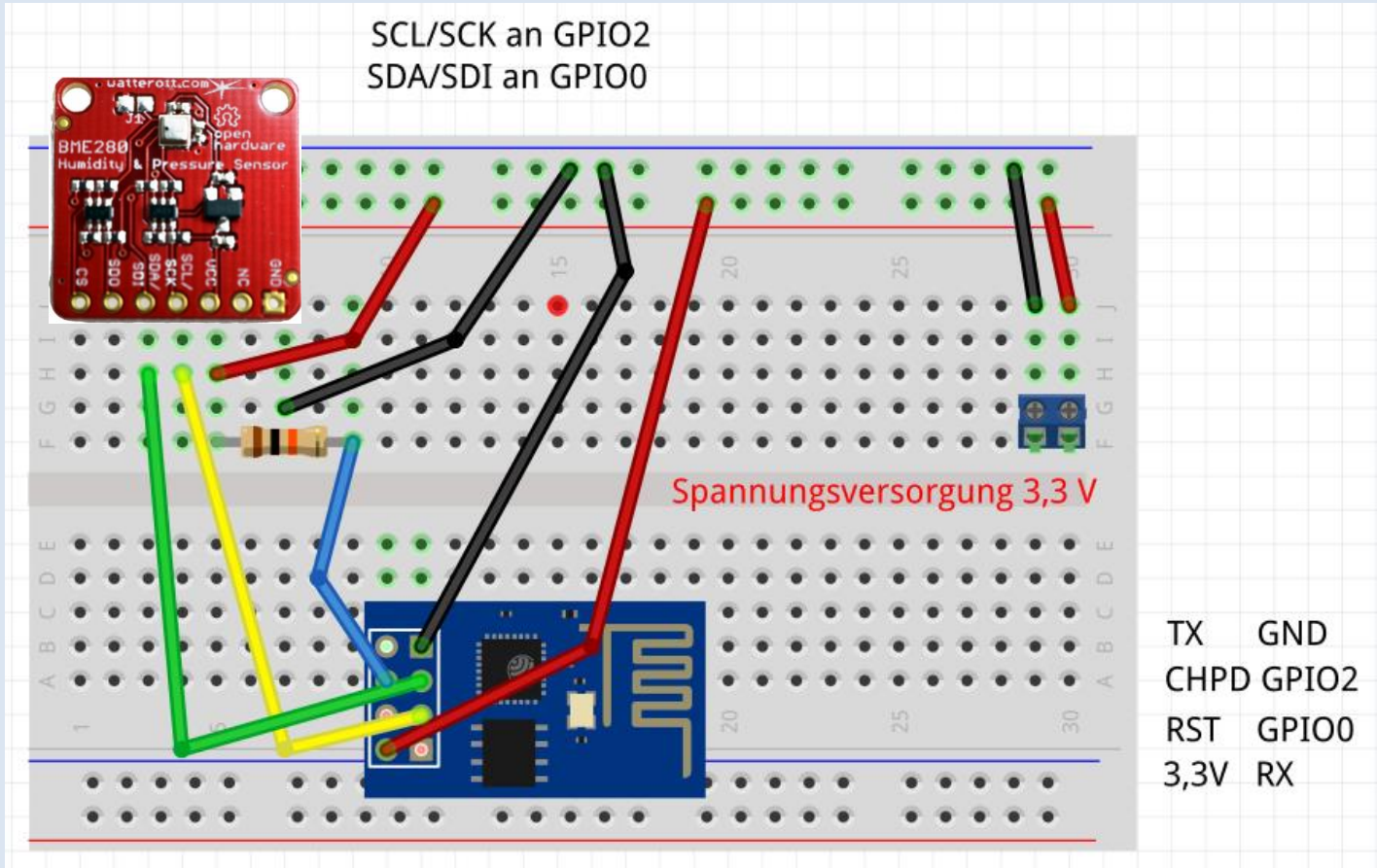
„**ESP-12E nodeMCU**“ in der I2C-Betriebsart; Foto



„**ESP-12E nodeMCU**“ & „bme280test.ino“ in der I2C-Betriebsart

| | |
|---|---|
| Wertvolle Tipps in | http://raphuscucullatus.blogspot.com/2017/07/chinesischer-bme280-sensor-und-esp8266.html |
| Sketch öffnen | Datei > Beispiele > „Adafruit BME280 Library“ > „**bme280test.ino**“ öffnen |
| Die Zeilen 23 bis 26 mit Kommentaren? | Da im I2C-Modus die SPI-DPIN-Zuordnung nicht benötigt wird, kann sie auskommentiert werden. |
| I2C in Zeile 30 ohne Kommentare | 30 *Adafruit_BME280 bme; // I2C* |
| I2C-Verkabelung „ESP-12E nodeMCU“ | `//I2C-wiring nodeMCU`<br>`//  BME280  nodeMCU ESP8266 12E`<br>`//  SCL/SCK  (Clock)  DPIN 1 GPIO5 Serial-Clock`<br>`//  SDA/SDI  (Serial Data In)  DPIN 2 GPIO4 Serial-Data (bi-directed)` |
| | Der Sketch „bme280test.ino“ sollte ohne Anpassungen funktionieren! |

„**ESP8266 ESP-01**“ in der I2C-Betriebsart; Schaltbild



„**ESP8266 ESP-01**“ & „bme280test.ino“ in der I2C-Betriebsart

| | | | |
|---|---|---|---|
| Wertvolle Tipps in | http://raphuscucullatus.blogspot.com/2017/07/chinesischer-bme280-sensor-und-esp8266.html | | |
| Die Zeilen 23 bis 26 mit Kommentaren? | Da im I2C-Modus die SPI-DPIN-Zuordnung nicht benötigt wird, kann sie auskommentiert werden. | | |
| I2C in Zeile 30 aktivieren | 30 *Adafruit_BME280 bme; // I2C* | | |
| | Board | SDA (data) | SCL (clock) |
| I2C-Verkabelung „ESP8266 ESP-01 “ | ESP-01 | GPIO0 | GPIO2 |
| Im Codeabschnitt **setup()** noch hinzufügen: | Wire.begin( 0, 2); | | |
| | | | |

| | |
|---|---|
| Link Weitere Infos: | github.com/watterott/BME280-Breakout |
| | Interessante Informationen, die für den Arduino-Programmierer zunächst nicht relevant sind! |
| Link:<br>Bosch BME280 | Produktinformation:<br>https://www.bosch-sensortec.com/en/bst/products/all_products/bme280 |
| | Relevante Informationen finden sich hier: |
| Link:<br>learn.watterott.com | http://learn.watterott.com/sensors/bme280/ |
| | Für den „BME280 MOD-1022 Weather Multi Sensor“. |
| | https://github.com/embeddedadventures/BME280 |
| | Für den Bosch BME280 Arduino/Teensy Library |
| | https://github.com/Protoinfy/BME280_Library |