1. Verbinde den Node mit dem PC.
2. Öffne Arduino IDE und das Programm: bme.ino
3. In den Voreinstellungen der Arduino IDE musst du folgende Internetadresse angeben:   
   Gehe zu Datei/Voreinstellungen/ im Feld Zusätzliche Bordverwalter-URLs:

<http://arduino.esp8266.com/stable/package_esp8266com_index.json>   
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

1. Lade über Werkzeuge/Boardverwalter die Hauptbibliotheken für den ESP8266 herunter:  
   Ein Bild, das Text enthält.

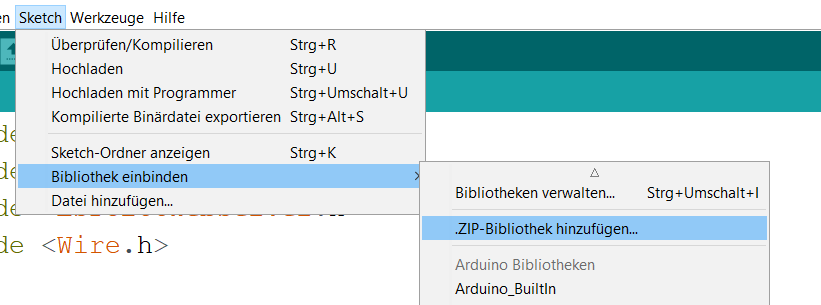
   Automatisch generierte Beschreibung
2. Wähle nun unter Werkzeuge folgendes Board herunter: NodeMCU 1.0 (ESP-12E Module)

Ein Bild, das Text enthält.

Automatisch generierte Beschreibungund vergewissere dich nochmals, dass der richtige Port ausgewählt wurde.

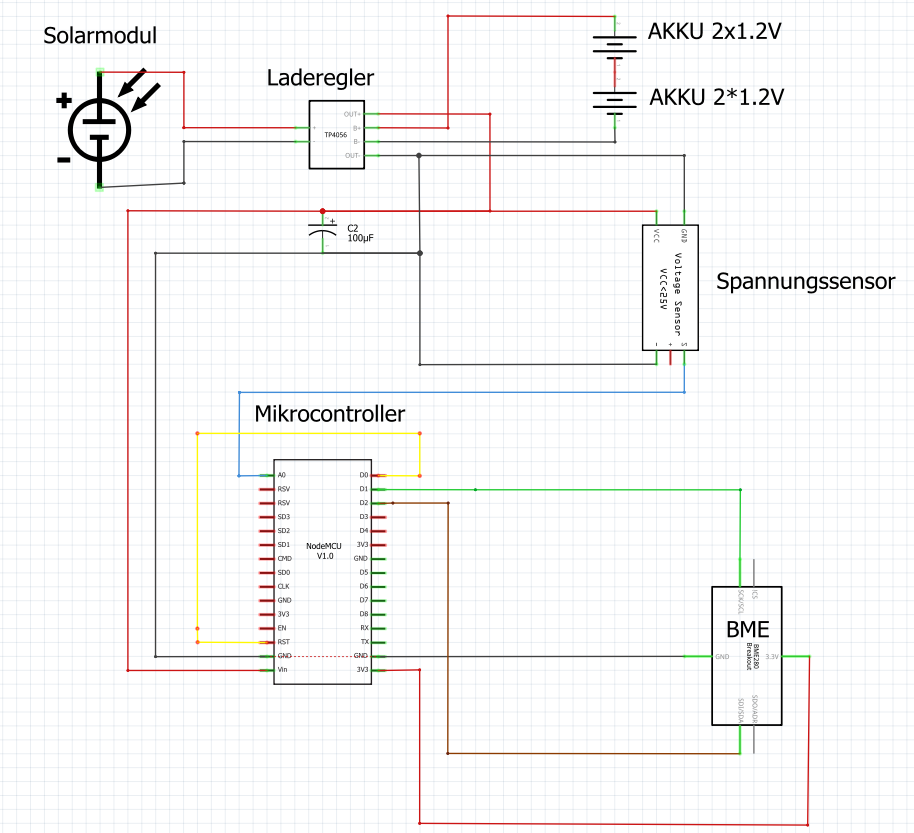
1. Trage in die bme.ino Datei deine WLAN Anmeldung ein: Ein Bild, das Text enthält.

   Automatisch generierte Beschreibung
2. Lade das Programm auf den Node. Wundere dich nicht, dass dies etwas länger als vermutet dauert und auch nicht, dass dann in roter Schrift kommentiert wird wieviel Prozent gerade hochgeladen sind.
3. Wenn das geklappt hat musst du am Node den RST Button klicken und einen Hardreset durchführen.
4. Öffnen nun am Rechner oder besser am Handy die IP Adresse die für deinen Node am Router vergeben wurde: z.B. 192.168.2.103 Verwende dazu Chrome, aber nicht Firefox!!!
5. Nun solltest du die Messwerte auf dem Handy angezeigt bekommen.
6. Toubleshooting: Fehlt eine der Bibliotheken, dann am besten bei google suchen (z.B. Adafruit\_BME280.h) und als zip runterladen und über Sketch/Bibliothek einbinden .ZIP-Bibliothek hinzufügen.



1. Bei den anderen Dateien meine Anmeldedaten überschreiben.
2. Auf die Uhr schauen!

Schaltplan:



Projektablauf: Bau einer Wetterstation

Anforderungen:

1. Die Wetterstation soll Wetterdaten erfassen: Temperatur, Luftdruck, Luftfeuchtigkeit, (Windgeschwindigkeit)
2. Die Energieversorgung soll möglichst autark (ohne Steckdose) gebaut sein und die Messung und Aufzeichnung auch über einen größeren Zeitraum gewährleisten.
3. Die Wetterdaten sollen gespeichert und graphisch dargestellt werden können, um diese mit anderen Wetterstationen vergleichen zu können.
4. Eine geographische Zuordnung soll den Vergleich mit anderen Wetterstationen ermöglichen und damit eventuell Wettertendenzen erkannt werden.
5. Der NodeMCU – ein Mikrocontroller mit WLAN-Modul

DeepSleep Modus:

WiFi.persistent( false ); // avoid writing to flash after every wakeup

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

while (!Serial);

Serial.println("Waking up...");

}

void loop() {

// put your main code here, to run repeatedly:

Serial.println("Looping...");

startDeepSleep();

}

void startDeepSleep(){

Serial.println("Going to deep sleep...");

ESP.deepSleep(5 \* 1000000);

yield();

}