

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

# WiFi Weather Station

În cadrul lecției de față vom realiza o stație meteo fără fir (cu conexiune WiFi).

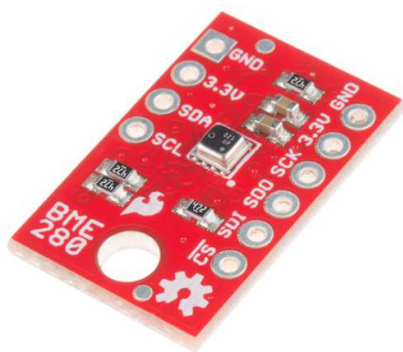
Proiectul se bazează pe senzorul Bosch BME280 – senzor integrat de mediu – capabil să măsoare presiunea atmosferică, temperatura în grade C și umiditatea aerului și să ofere, prin intermediul funcțiilor bibliotecii software, temperatura în grade F, altitudinea în metri / picioare. Cu alte cuvinte, acest senzor este o mică stație meteo într-un singur circuit integrat fiind unul dintre cele mai avansate circuite de acest tip.



[https://www.bosch-sensortec.com/bst/products/all\\_products/bme280](https://www.bosch-sensortec.com/bst/products/all_products/bme280)

Având în vedere formatul extrem de mic (2.5mm x 2.5mm) pentru a putea utiliza acest circuit în cadrul montajului nostru vom folosi modulul Sparkfun Breakout BME280. Senzorul poate fi conectat la o placă de dezvoltare utilizând magistrala I2C sau SPI (la alegere) și funcționează la tensiunea de alimentare de 3.3V – nu suportă niveluri logice de 5V. Dacă dorim să utilizăm acest senzor cu o placă de dezvoltare ce funcționează la 5V (Arduino Uno, Arduino Mega, Arduino Leonardo) este necesară utilizarea unui convertor de nivel (3.3V-5V):

<https://www.robofun.ro/electronice/generale/logic-level-converter-bi-directional> .



<https://www.robofun.ro/senzori/vreme/sparkfun-atmospheric-sensor-breakout-bme280>

<https://www.robofun.ro/forum/>

Senzorul poate fi utilizat fără probleme cu plăci de dezvoltare ce funcționează la 3.3V, câteva exemple:

- Arduino Pro Mini 328 - 3.3V/8MHz  
[https://www.robofun.ro/arduino/arduino\\_pro\\_mini\\_328\\_8mhz](https://www.robofun.ro/arduino/arduino_pro_mini_328_8mhz)
- Pro Micro 3.3V/8MHz - ATmega 32U4  
<https://www.robofun.ro/arduino/pro-micro-3-3v-8mhz>
- Arduino Due  
<https://www.robofun.ro/arduino/arduino-due>
- Arduino Industrial - Olimexino 328  
[https://www.robofun.ro/platforme/arduino\\_dev/arduino-industrial](https://www.robofun.ro/platforme/arduino_dev/arduino-industrial)

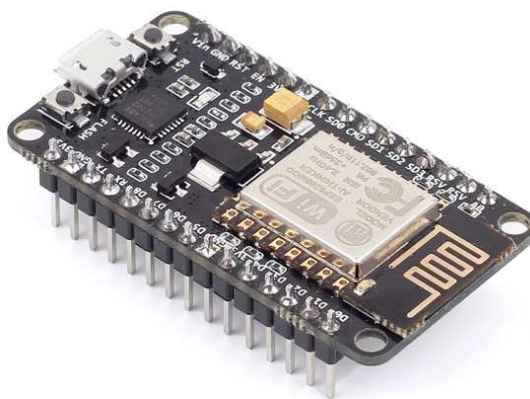
Pentru mai multe detalii legate de funcționarea modului SparkFun BME280 Breakout se poate parcurge următorul Hookup Guide:

<https://learn.sparkfun.com/tutorials/sparkfun-bme280-breakout-hookup-guide>

Utilizarea în program a senzorului este facilitată de biblioteca software Sparkfun BME280 Arduino Library ce trebuie instalată sub mediul Arduino IDE înainte de scrierea / încărcarea programului:

[https://github.com/sparkfun/SparkFun\\_BME280\\_Arduino\\_Library](https://github.com/sparkfun/SparkFun_BME280_Arduino_Library)

Pentru implementarea proiectului vom utiliza o placă de dezvoltare bazată pe controlerul WiFi ESP8266 și anume placa NodeMCU. Chiar dacă nu este o placă din familia Arduino această placă poate fi programată cu ajutorul mediului de dezvoltare Arduino IDE și majoritatea bibliotecilor software funcționează fără probleme – inclusiv Sparkfun BME280 Arduino Library. Placa funcționează la 3.3V și se poate alimenta prin intermediul conexiunii USB. Avantajul major al acestei plăci de dezvoltare este, bineînțeles, conectivitatea de rețea WiFi – scutind necesitatea utilizării unui shield specializat.



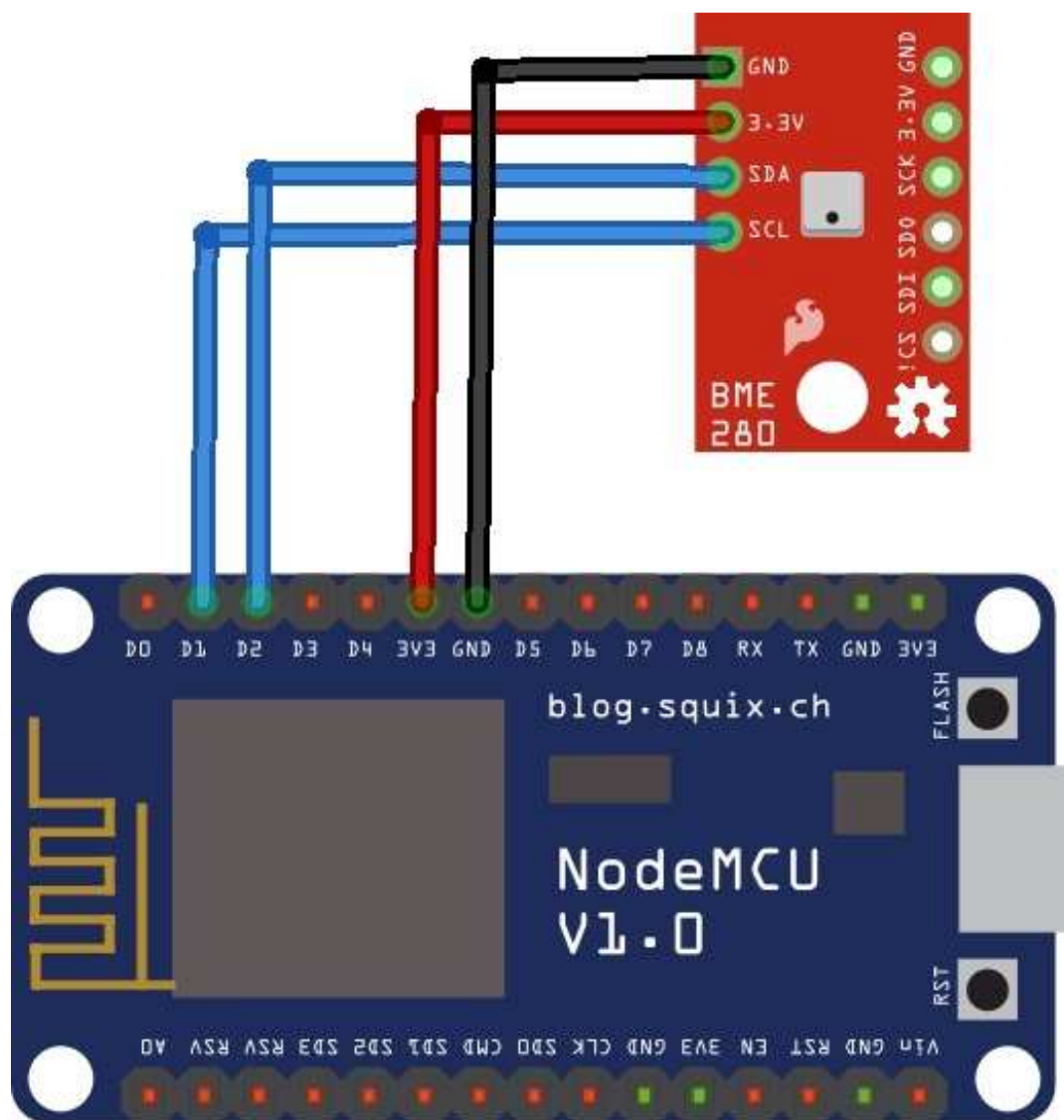
<https://www.robofun.ro/wireless/wireless-wifi/NodeMCUv2-ESP8266>

Pentru configurarea mediului Arduino IDE se pot urma pașii descriși în materialul "Quick Start to Nodemcu (ESP8266) on Arduino IDE":

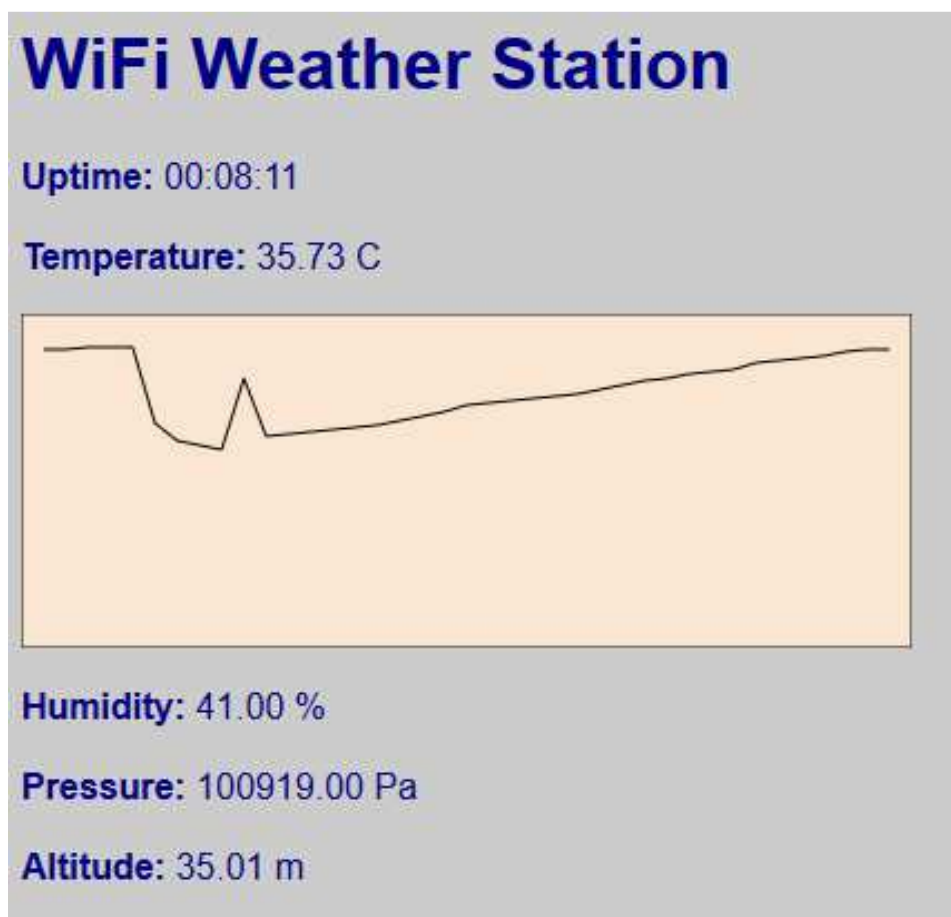
<http://www.instructables.com/id/Quick-Start-to-Nodemcu-ESP8266-on-Arduino-IDE/>

Conectarea între senzorul BME280 și placa de dezvoltare se va realiza prin intermediul magistralei I2C (conform schemei de interconectare de mai jos):

- Pinul SCL al senzorului se va conecta la pinul D1 al plăcii de dezvoltare;
- Pinul SDA al senzorului se va conecta la pinul D2 al plăcii de dezvoltare;
- Pinul de GND la GND și pinul de 3.3V la 3V3.



Scopul sistemului este acela de a face disponibile informațiile provenite de la senzorul BME280 prin intermediul unei interfețe web (accesibilă de pe orice sistem, PC sau telefon mobil inteligent. din aceeași rețea locală ca și sistemul meteo). Interfața web va afișa timpul de funcționare a sistemului (Uptime), temperatura în grade C (Temperature), un grafic al evoluției temperaturii, umiditatea (Humidity), presiunea atmosferică (Pressure) și altitudinea (Altitude).



Accesarea interfeței web se poate face cu ajutorul unui client web (browser) accesând adresa IP a sistemului meteo – adresa se preia dinamic (prin DHCP) de către sistem și este afișată în Serial Monitor la pornirea sistemului.

COM26

```
FUEÚOP...@AieI"WMQ'êKY@afI]_[5"000ÄTÖ00aa\%00!  
Connected to [REDACTED]  
IP address: 192.168.100.7  
HTTP server started  
Starting BME280... result of .begin(): 0x60
```

Programul sistemului se bazează pe exemplul *AdvancedWebServer* din secțiunea *Examples/ESP8266WebServer*. În exemplu se înserează partea de achiziție parametrii de la senzorul BME280:

```
#include <stdint.h>
#include "SparkFunBME280.h"
#include "Wire.h"

BME280 mySensor;

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
```

În program trebuie personalizate datele de conectare la rețeaua WiFi:

```
const char *ssid = "....";
const char *password = "....";

ESP8266WebServer server ( 80 );
```

Led-ul de pe placa de dezvoltare NodeMCU nu este conectat la pinul 13 ci la pinul 0. Mai mult de cât atât, comanda de aprinderea a acestuia se face pe "0" nu pe "1". În cadrul sistemului meteo led-ul de pe placa de dezvoltare se va aprinde la conectarea unui client web.

```
const int led = D0;
```

Vectorul *temperature\_history* va stoca valorile de temperatura ce vor folosi la generarea graficului de evoluție.

```
float temperature_history [39];
int temperature_index = 0;

char temp[600];
```

Procedura *handleRoot()* va construi pagina web la conectarea unui client.

```
void handleRoot() {
    digitalWrite ( led, 0 );
    int sec = millis() / 1000;
    int min = sec / 60;
    int hr = min / 60;
    temperature_history[temperature_index] = mySensor.readTempC();
    String t = String(temperature_history[temperature_index],2);
    Serial.println(t);
    temperature_index ++;
    if (temperature_index > 38) temperature_index = 0;
    String p = String(mySensor.readFloatPressure(),2);
    Serial.println(p);
    String a = String(mySensor.readFloatAltitudeMeters(),2);
```

```

Serial.println(a);
String h = String(mySensor.readFloatHumidity(),2);
Serial.println(h);
snprintf ( temp, 600,"%02d:%02d:%02d", hr, (min % 60), (sec % 60));
String page = "<html>\
<head>\
<meta http-equiv='refresh' content='10'/>\
<title>WiFi Weather Station</title>\
<style>\
    body { background-color: #cccccc; font-family: Arial, Helvetica,
Sans-Serif; Color: #000088; }\
</style>\
</head>\
<body>\
    <h1>WiFi Weather Station</h1>\
    <p><b>Uptime:</b> " + String(temp) + " </p>\
    <p><b>Temperature:</b> " + t + " C </p>\
    <p><img src=\"/test.svg\" /></p>\
    <p><b>Humidity:</b> " + h + " %</p>\
    <p><b>Pressure:</b> " + p + " Pa</p>\
    <p><b>Altitude:</b> " + a + " m</p>\
</body>\
</html>";
page.toCharArray(temp,page.length()+1);
server.send ( 200, "text/html", temp );
digitalWrite ( led, 1 );
}

```

În secțiunea *setup()* se va realiza conexiunea la rețeaua WiFi specificată, se va porni și configura conexiunea web pentru deservirea solicitărilor client și se va realiza partea de inițializare a senzorului BME280.

```

void setup()
{
    Serial.begin(57600);

    pinMode ( led, OUTPUT );
    digitalWrite ( led, 1 );

    WiFi.begin ( ssid, password );

    while ( WiFi.status() != WL_CONNECTED ) {
        delay ( 500 );
        Serial.print ( "." );
    }

    Serial.println ( "" );
    Serial.print ( "Connected to " );
    Serial.println ( ssid );
    Serial.print ( "IP address: " );
    Serial.println ( WiFi.localIP() );

    server.on ( "/", handleRoot );
    server.on ( "/test.svg", drawGraph );
    server.on ( "/inline", []() {

```

```

    server.send ( 200, "text/plain", "this works as well" );
} );
server.begin();
Serial.println ( "HTTP server started" );

mySensor.settings.commInterface = I2C_MODE;
mySensor.settings.I2CAddress = 0x77;
mySensor.settings.runMode = 3; //Normal mode
mySensor.settings.tStandby = 0;
mySensor.settings.filter = 0;
mySensor.settings.tempOverSample = 1;
mySensor.settings.pressOverSample = 1;
mySensor.settings.humidOverSample = 1;

Serial.print("Starting BME280... result of .begin(): 0x");
delay(10);
Serial.println(mySensor.begin(), HEX);
}

```

În cadrul secțiunii *loop()* se vor trata solicitările de conectare client:

```
void loop() { server.handleClient(); }
```

Procedura *drawGraph()* realizează graficul evoluției temperaturii:

```

void drawGraph() {
    String out = "";
    char temp[100];
    out += "<svg xmlns=\"http://www.w3.org/2000/svg\" version=\"1.1\" "
width=\"400\" height=\"150\">\n";
    out += "<rect width=\"400\" height=\"150\" fill=\"rgb(250, 230, 210)\" "
stroke-width=\"1\" stroke=\"rgb(0, 0, 0)\" />\n";
    out += "<g stroke=\"black\">\n";
    int y = (32 - temperature_history[0])*15;
    for (int x = 10; x < 390; x+= 10) {
        int y2 = (32 - temperature_history[x/10])*15;
        sprintf(temp, "<line x1=\"%d\" y1=\"%d\" x2=\"%d\" y2=\"%d\" stroke-
width=\"1\" />\n", x, 70 + y, x + 10, 70 + y2);
        out += temp;
        y = y2;
    }
    out += "</g>\n</svg>\n";
    server.send ( 200, "image/svg+xml", out);
}

```

Programul a fost realizat și testat cu Arduino IDE 1.6.9, extensia ESP8266 Community 2.3.0 și biblioteca Sparkfun BME280 1.0.0.

Proiectul a fost realizat cu ajutorul unui fundal sonor asigurat de albumul [Ozzy Osbourne – No More Tears](#).



Programul poate fi completat foarte ușor cu calculul și afișarea mai multor parametrii (punct de rouă, temperatură resimțită) și chiar cu realizarea unei predicții de evoluție a vremii (pe baza evoluției în timp a presiunii atmosferice).

Punct de rouă

[https://ro.wikipedia.org/wiki/Punct\\_de\\_rou%C4%83](https://ro.wikipedia.org/wiki/Punct_de_rou%C4%83)

What is the AccuWeather RealFeel Temperature?

<http://www.accuweather.com/en/weather-news/what-is-accuweather-realfeel/7198202>

Weather forecasting

[https://en.wikipedia.org/wiki/Weather\\_forecasting](https://en.wikipedia.org/wiki/Weather_forecasting)