

# Sistem de interpretare a stării atmosferice - Iordache Alexandra 332AA

## Introducere

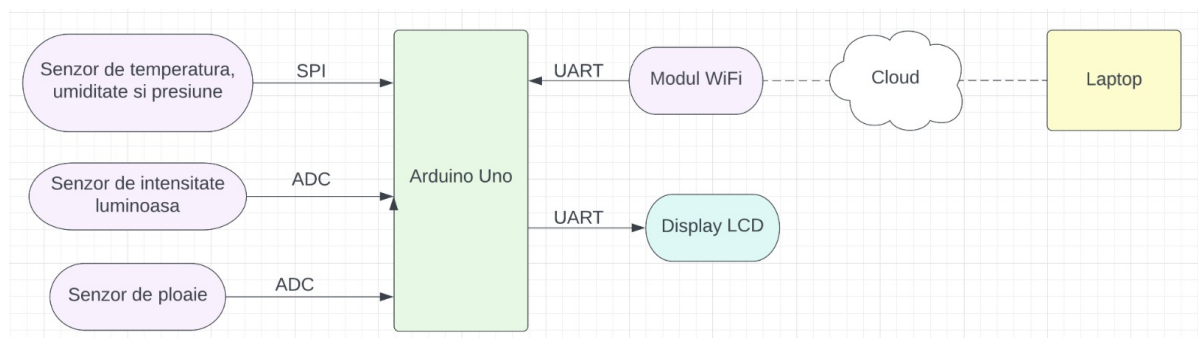
Sistemul de interpretare a stării atmosferice (senin, înnorat, ploaie, ninsoare) are ca scop principal monitorizarea vremii în timp real. Acesta reprezintă este un set de tehnologii și instrumente care permit colectarea, analizarea și interpretarea datelor meteorologice pentru a furniza informații despre starea actuală și viitoare a vremii. Sistemul propus utilizează IoT pentru a face datele sale ușor accesibile în timp real.

Sistemul de interpretare a vremii este util pentru ca poate furniza informații utile despre starea actuală și viitoare a vremii, ceea ce poate fi folositor în diverse activități umane.

## Descriere generală

## Schema bloc

Mai jos este ilustrată schema bloc care cuprinde modulele acestui proiect:



## Modul de funcționare

Proiectul se ocupa de monitorizarea schimbărilor meteorologice și climatice, urmărind, prin intermediul unor senzori, evoluția valorilor temperaturii, a umidității, a presiunii și a intensității luminii. Aceste senzori trimit datele către pagina web, iar datele senzorului sunt reprezentate grafic sub forma de statistici. În centrul sistemului este prezentă plăcuța Arduino Uno care comunica cu toate celelalte componente. La ea sunt legați cei 3 senzori, modulul Wifi, cât și display-ul LCD.

Datele încărcate pot fi ușor folosite ca referințe viitoare sau comparate cu intervale obținute din alte surse meteorologice asemănătoare. Sistemul încorporat este o integrare a dispozitivelor senzoriale, comunicare fără fir, care permite utilizatorului să acceseze de la distanță diferiți parametri și să stocheze datele.

Pentru implementarea proiectului, am aplicat notiuni din următoarele laboratoare:

- Laboratorul 1: UART
- Laboratorul 4: ADC
- Laboratorul 5: SPI

## Hardware Design

Componentele folosite în cadrul acestui proiect sunt:

- Placuta Arduino Uno
- Modul Sensor Barometric de presiune BME280
- Modul cu Fotorezistor
- Modul Sensor de Ploaie
- Modul WiFi ESP-01
- Display LCD

## Software Design

Proiectul a fost realizat în ARDUINO IDE, am folosit:

- biblioteca Adafruit\_BMP085 pentru a lua structura senzorului de presiune și temperatura BMP.
- biblioteca Wire pentru a putea comunica prin protocolul I2C.
- biblioteca Adafruit\_Sensor pentru a avea o interfață comună pentru senzorii de la Adafruit.
- biblioteca LiquidCrystal pentru a controla și comunica cu afișaje cu cristale lichide (LCD)

Am folosit și un modul WiFi ESP8266 pentru a putea comunica cu Cloud-ul pus la dispoziție de ThingSpeak. Astfel, am conectat pinul de transmisie al modulului la pinul 2 al plăcuței Arduino și pinul de recepție al modulului la pinul 3. Am stocat în 4 variabile:

- numele rețelei (SSID-ul) la care dispozitivul Arduino sau microcontrolerul se va conecta
- parola pentru rețeaua WiFi
- cheia API utilizată pentru autentificarea și accesul la serviciul cloud ThingSpeak
- adresa serverului către care dispozitivul se va conecta pentru a trimite date către serviciul ThingSpeak
- numărul portului utilizat pentru comunicarea cu serverul

În funcția de setup initializez seriala și verific dacă adresa I2C a senzorului BME280 este cea corectă. În caz contrar, se afișează pe seriale un mesaj de eroare. De asemenea, verific și conexiunea cu modulul WiFi prin comanda AT, setez modul de funcționare al modulului în modul stație

(AT+CWMODE=1) și trimite SSID-ul și parola pentru rețeaua de WiFi modulului pentru a putea realiza conexiunea.

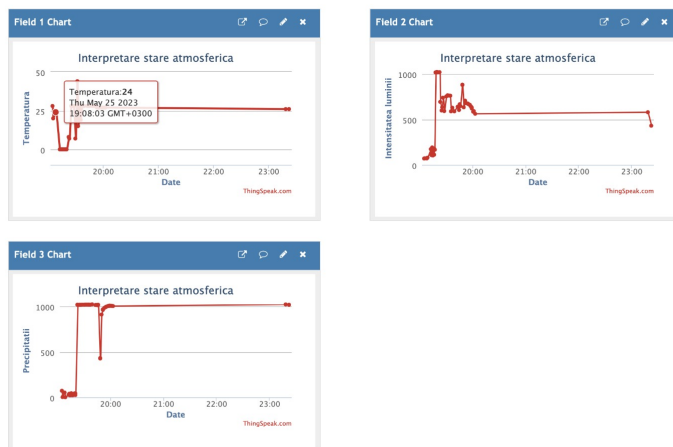
În funcția de loop apelez cele 3 funcții specifice fiecărui senzor folosit:

- `bme.readTemperature()`, `bme.readPressure()`, `bme.readAltitude`, `bme.readHumidity()` pentru a citi valorile de temperatura, presiune, altitudine și umiditate primite de la senzorul BME280. Valoarea temperaturii este afișată atât pe seriala, cât și pe display-ul LCD, dar este și trimisă ca parametru la platforma de Cloud.
- Pentru a citi valoarea senzorului de ploaie, în primul rând am configurat pinul A0 pentru citirea analogică și am setat tensiunea de referință la 5. Ulterior am selectat portul de intrare pe care se va face conversia, urmând să se înceapă conversia. După ce procesul de conversie s-a terminat, rezultatul obținut este trimis afișat pe seriala și trimis ca parametru la platforma de Cloud.
- Pentru a citi valoarea modulului cu fotorezistor, am configurat pinul A1 pentru citirea analogică, iar rezultatul obținut este trimis afișat pe seriala și trimis ca parametru la platforma de Cloud.

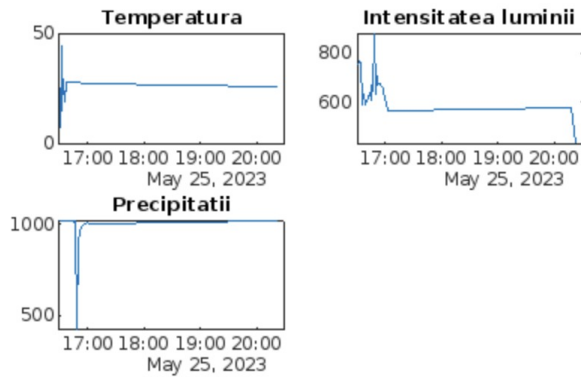
## Rezultate Obținute

Datele obținute de la senzori sunt transmise printr-o serie de comenzi în Cloud-ul pus la dispoziție de ThingSpeak. Prin intermediul a 3 grafice se poate vizualiza în timp real evoluția valorilor pentru temperatura, intensitatea luminii și a precipitațiilor.

Last entry: about 20 hours ago  
Entries: 58



Am realizat și o analiză a datelor primite pentru a stabili starea atmosferică. Codul preia ultimele 10 valori din fiecare grafic menționat mai sus, realizează o medie a acestor valori și afișează pe ecran starea atmosferică în funcție de rezultatele obținute.



Exista și posibilitatea de a vizualiza grafic evoluția în timp a datelor obținute:

"NOT RAINING"

"It is slightly bright outside"

"It is sunny outside!"

De asemenea, am realizat și o analiza comparativa cu un alt sistem meteo. Tot prin intermediul ThingSpeak-ului, am preluat un canal al unei stații meteo care afișează în timp real temperatura și umiditatea din București. Astfel, cu datele de la acest canal împreună cu datele primite de la senzorii mei am realizat un grafic pentru a se poate observa diferența între valorile temperaturii obținute în cele 2 cazuri. Datele nu sunt identice, dar sunt apropiate, întrucât senzorii folosiți în cadrul acestui proiect nu sunt cei mai performanti astfel ca pot apărea erori la preluarea datelor.

## Concluzii

A fost un proiect interesant de realizat, întrucât a venit cu numeroase provocări, din fiecare reușind sa învăț ceva nou. Am realizat ce înseamnă sa lucrezi în acest domeniu de embedded, începând cu citirea datasheet-ului fiecărei componente utilizate, pana la realizarea montajului și crearea codului din spate care sa ofere o logica partii hardware. Proiectului de fata i se pot adauga mai multe funcționalități, aceasta varianta fiind una de baza, dar folositoare pentru o dezvoltare ulterioară.

## Download

[pm\\_alex\\_a\\_iordache.zip](#)

## Jurnal

- 13 martie - alegere tema proiect
- 21 martie - achiziția unor componente
- 18 mai - achiziția restului de componente
- 22 mai - finalizare hardware + software

## Bibliografie/Resurse

- <https://electrosome.com/interfacing-rain-sensor-arduino/>
- <https://startingelectronics.org/tutorials/arduino/modules/photo-resistor/>
- <https://randomnerdtutorials.com/bme280-sensor-arduino-pressure-temperature-humidity/>
- <https://html.alldatasheet.com/html-pdf/1132060/BOSCH/BME280/3773/21/BME280.html>
- <https://www.youtube.com/watch?v=764bRMvGZR4>
- <https://www.deviceplus.com/arduino/how-to-connect-your-arduino-to-wi-fi/>
- <https://www.elprocus.com/esp8266-wi-fi-module/>
- <https://ocw.cs.pub.ro/courses/pm>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/adarmaz/detectare-stare-atmosferica>



Last update: **2023/05/26 18:21**