

# Air Quality Monitor

## Autor

[Teodor Szente](#)

## Introducere

Proiectul constă în dezvoltarea unui sistem de monitorizare a calității aerului. Datele vor fi colectate într-o interfață grafică, unde vor exista și predicții despre evoluția calității aerului. Predicțiile se vor baza pe pattern-urile observate în măsurătorile făcute periodic pentru o durată lungă de timp și îl vor ajuta pe utilizator să se ferească de aerul poluat.

Consider ca un astfel de dispozitiv este util într-un oraș tot mai poluat ca București.

## Descriere generală

Dispozitivul va conține doi senzori. **Sharp GP2Y1014AU0F Dust Sensor** pentru detectarea impurităților (prafului) din aer.

Aceste date vor fi colectate de către MCU și trimise prin HTTP la un server folosind un modul ESP8266.

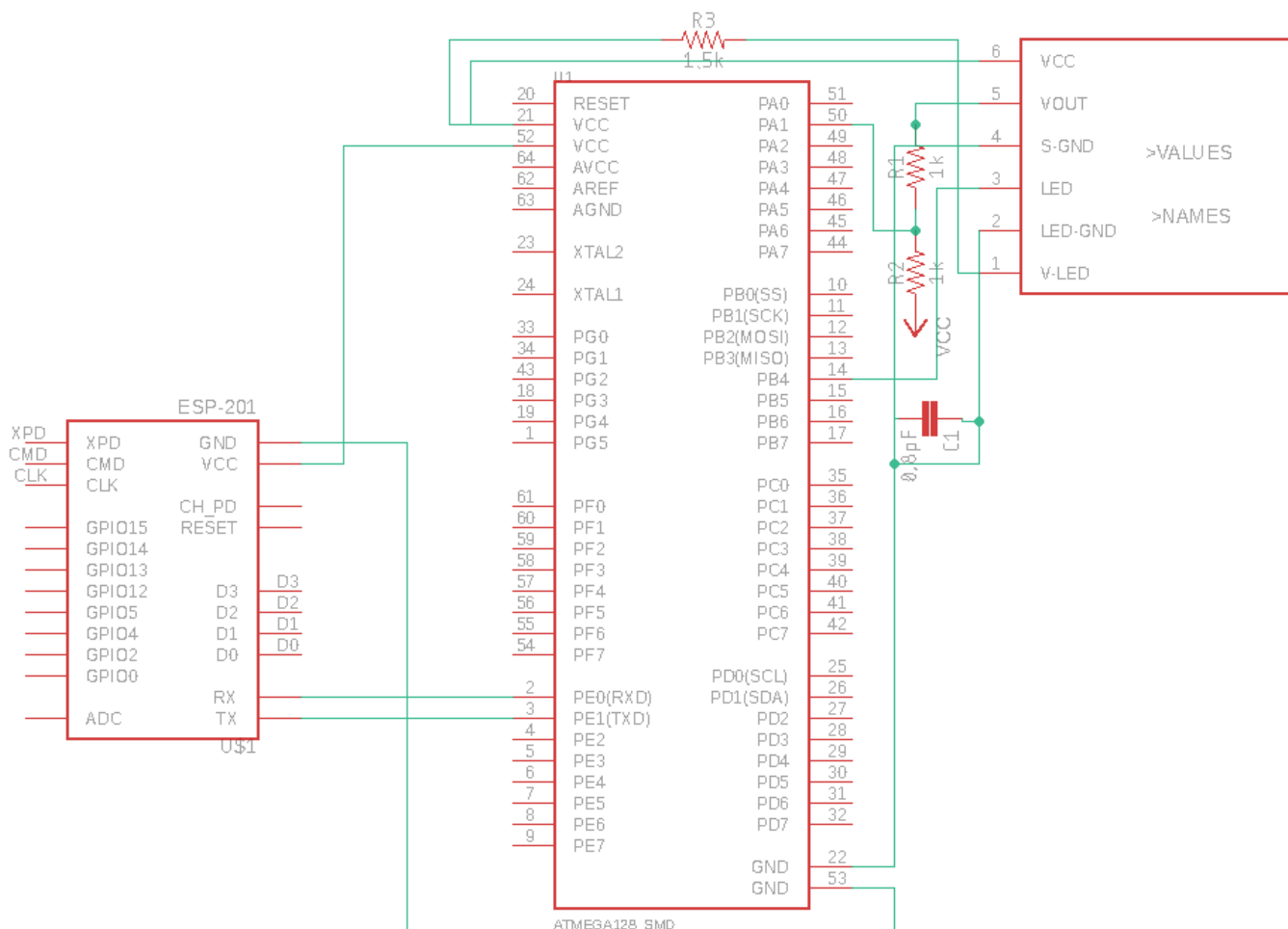


## Hardware Design

Lista de piese:

- Placuta PM
- [Sharp GP2Y1014AU0F Dust Sensor](#)
- [ESP8266 ESP-01 WiFi Serial Transceiver](#)

Senzorul GP2Y1014AU0F funcționează la 5V dar modulul de wifi la 3.3 așa ca am ales merg pe 3.3V și să fac un divizor de tensiune pe ieșirea senzorului. Suplimentar am incorporat și o rezistență și un tranzistor [necesare](#) senzorului.



## Software Design

Dezvoltarea software-ului s-a despărțit în 3 componente: realizarea codului pentru microcontroller, realizarea web server și realizarea interfeței grafice.

## Microcontroller

### Programarea ESP8266

Modulul ESP8266 este conectat prin serial, comunicația se face prin USART0, microcontroller transmite către acestea comenzi de AT. Nu am găsit tutoriale de ESP+USART pentru Atmega324p deoarece majoritatea sunt create pentru Atmega128. Am apelat la ajutor din proiectele anterioare și am reușit să trimit primul pachet prin Wifi.

## Colectarea datelor

Scopul device-ului este de a monitoriza aerul deci acesta va citi la intervale regulate de timp datele de la senzorul de praf. Folosind exemplele/documentația din laboratorul de 5 [pentru a citi analogic](#) de pe pinul PA1 prin prin \*long-polling\*.

## Backend

Datele venite de la device trebuiesc așa că am ales să creez un server în node js și să țin datele în mongodb folosind mongoose. Modelul ales pentru a reprezenta o intrare este:

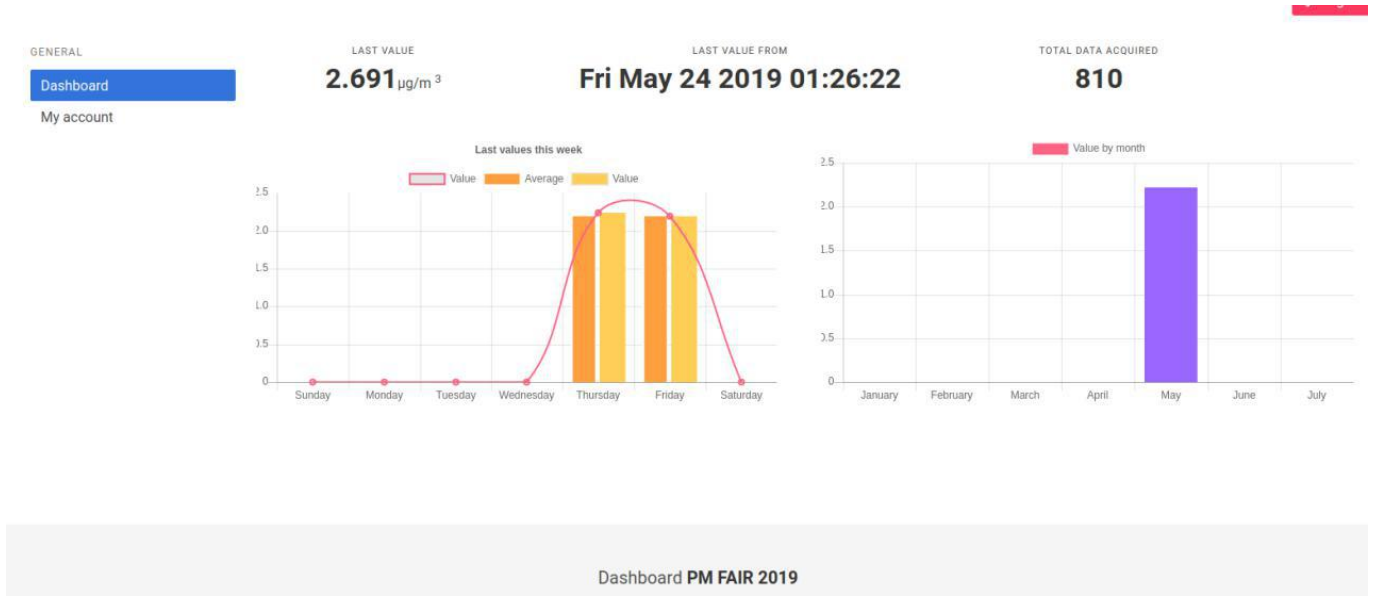
```
var entriesSchema = mongoose.Schema({
  value: {
    type: Number,
    required: true,
  },
  date: {
    type: Date, default: Date.now
  }
});
```

Codul pentru query-urile peste intrari poate fi consultat [aici](#)

## UI

UI-ul este construit folosind [Char.js](#) și [Semantic](#).

## Rezultate Obținute



## Concluzii

A fost un proiect interesant. Dacă aş alege din nou aş opta pentru un modul ESP32 pe care aş scrie Espruino/Micropython. Probabil pentru un proiect real aş investi mai mult in sensorul de calitatea a aerului astfel încat să detecteze orice fel de impuritate și alte gaze precum CO2.

## Download

[teodor\\_szente.zip](#)

## Jurnal

- 20 aprilie - Adaugare sectiuni: Introducere, Descriere Generala, etc.
- 10 mai - Finalizare placuta PM (lipit componente, incarcat bootloader).
- 15 mai - Ridicare comanda cu toate piesele necesare.
- 20 mai - Realizare comunicare prin ESP.
- 22 mai - Citire date de la senzor.
- 23 mai - Implementare interfață grafică.

## Bibliografie/Resurse

- [ESP8266 WiFi Module Interface with AVR ATmega16](#)
- [Application Guide for Sharp](#)

- [The USART of AVR](#)
- [Laborator 6](#)
- [Remote pet feeder](#)
- [Această pagină în format PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2019/amocanu/air-quality-monitor>

Last update: **2021/04/14 15:07**

