

Room Monitor

Introducere

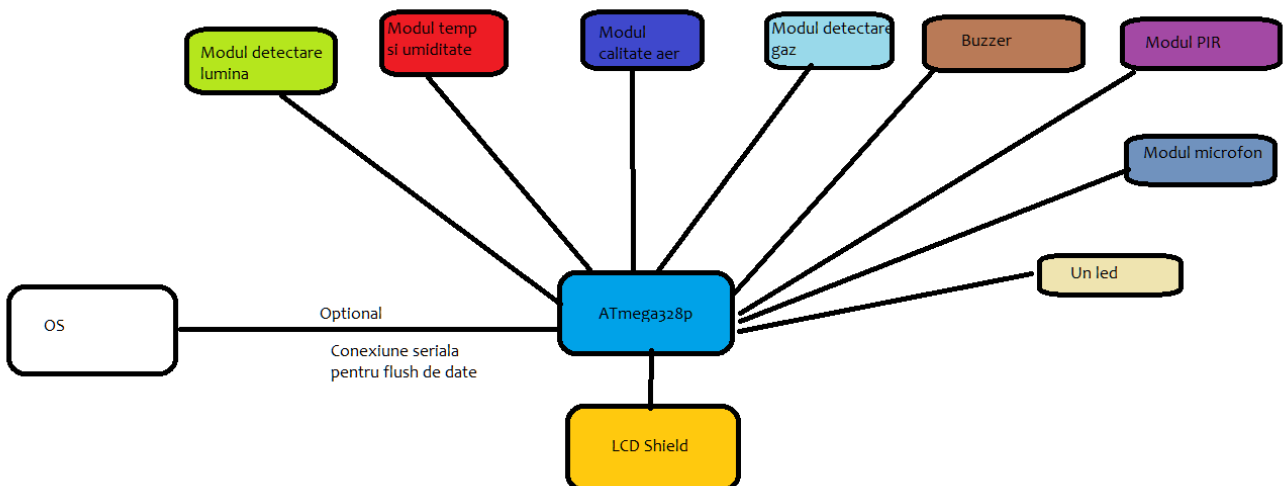
Room monitor confera utilizatorului detalii despre starea camerei in care se afla.

Aceste detalii includ:

- temperatura si umiditatea din camera
- calitatea aerului
- intensitatea luminii
- nivelul de zgomot
- daca exista scurgeri de gaz sau nu

In aceasta perioada, cand multi oameni lucreaza de acasa, conditiile din camera de lucru au devenit si mai importante pentru sanatate.

Descriere generală



Initial, pentru 2 minute, se incalzesc senzorul de gaz si cel de calitate a aerului.

Apoi se face prima citire a senzorilor, iar datele sunt afisate pe lcd.

Pentru a economisi curent, lcd-ul se va inchide daca nu este detectata miscare in proximitate

Senzorul PIR si cel de gaz sunt consultati la fiecare secunda, data fiind importanta lor.

Ceilalti sensori sunt consultati la fiecare 5 minute, cand se face(optional) si flush pe seriala a datelor.

Hardware Design

Lista componente

1. [Sensor PIR](#)
2. [Modul microfon](#)
3. [Buzzer](#)
4. [Led catod comun](#)
5. [Modul intensitate luminoasa](#)
6. [Modul detectare gaz](#)
7. [Modul detectare calitate aer](#)
8. [Senzor umiditate si temperatura](#)
9. [Shield lcd](#)
10. Fire tata-tata, mama-tata, breadboard, rezistente.

Software Design

Codul arduino a fost dezvoltat in [Arduino IDE](#)

Pe langa codul pentru microcontroler, am scris si un mic script in python care preia datele de pe seriala si le scrie intr-un fisier.

Biblioteci

Am folosit in dezvoltarea pe microcontroller urmatoarele biblioteci third-party:

1. [Arduino LiquidCrystal Library](#)
2. [Arduino DHT Sensor Library](#)

Pentru "consumerul" de seriala, am folosit urmatoarele biblioteci python:

1. [Python serial](#)
2. [Python datetime](#)
3. [Python calendar](#)

Structuri de date

Structura principala este numita room state si are urmatoare definitie:

```
typedef struct roomstate {
    float temperature, humidity, heatIndex;
    int airQuality;
    int airQualityPercentage;
    int lightIntensity;
    int gasValue;
    int motion;
    int sound;
} RoomState;
```

Aici este tinuta starea camerei, la un moment dat.

Funcții implementate

Funcțiile principale sunt următoarele:

```
void initSensors(); // Aici, timp de 2 secunde, se incalzesc cei 2 sensori
de aer.
void flushRoomState(); // Aici structura de date principala este trimisa pe
seriala.
void handleSensors(); // Un wrapper peste un agregat de functii mai mici
care citesc fiecare senzor in parte.
void buzz(float seconds, int sign); // Functie pentru buzz.
void handleLCDScreen(); // Functie care imi afiseaza starea la lcd.
void printLCDDateTime(); // Functie care afiseaza pe lcd data si ora.
void printLCDTemperature(); // Functie care afiseaza pe lcd informatii
despre temperatura.
void printLCDAirQuality(); // Functie care afiseaza pe lcd informatii despre
calitatea aerului.
void loop(); // Citim sensorii, daca s-au implinit 5 minute, atunci dam
flush.
```

Pe raspberry pi , daemon-ul care comunica cu seriala este pornit la boot (am modificat fisierul /etc/rc.local)

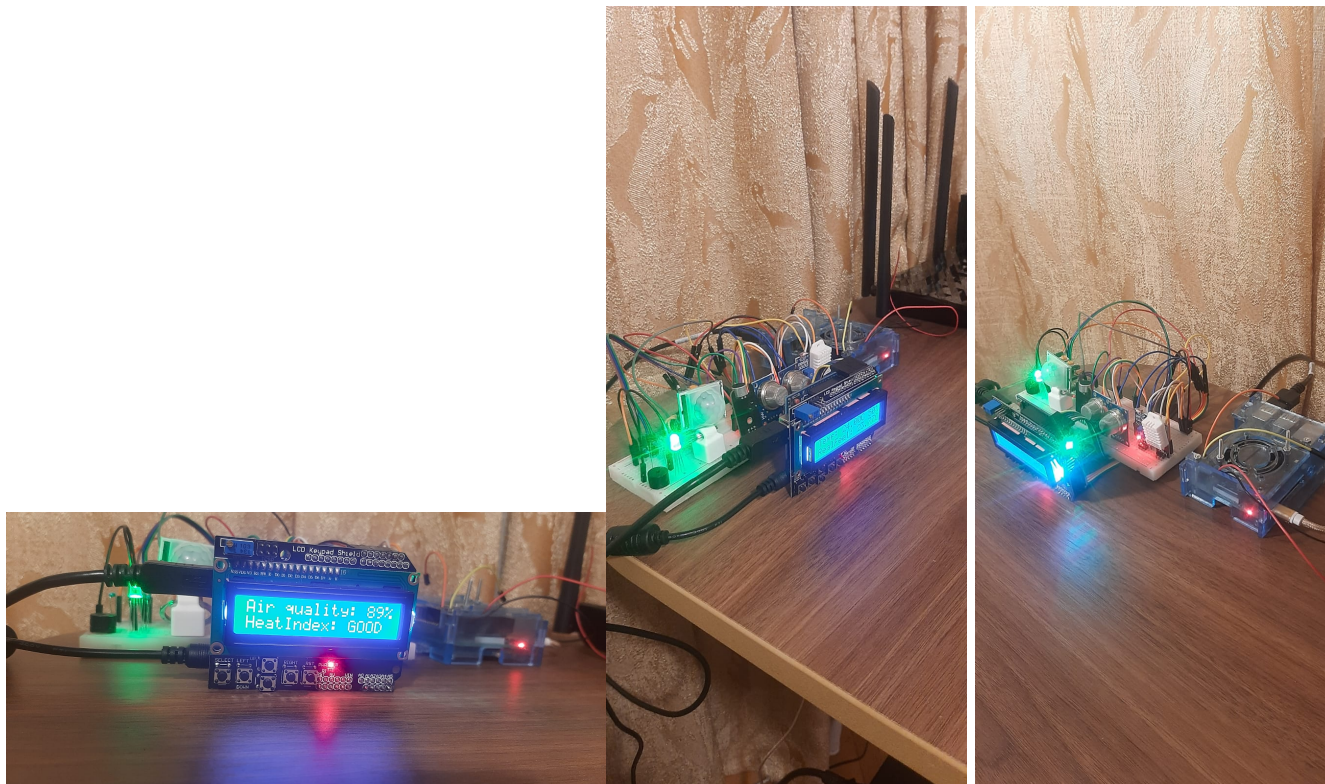
Codul python are urmatoarea structura:

```
ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1) // deschidem
comunicarea
f = open("./usb/data", "a") // deschidem si fisierul pentru stocare

while True:
    if ser.in_waiting > 0:
        line = ser.readline().decode('utf-8').rstrip()
        if line == "getHour":
            // microcontrollerul cere ora
        if line == "getDate":
            // microcontrollerul cere data
```

```
if line == "flush":  
    // microcontrollerul doreste sa ne dea datele.
```

Rezultate obtinute



Concluzii

Mi-a facut placere sa implementez acest proiect, deoarece am construit ceva care va ramane, pe care il folosesc. O sa continui dezvoltarea, incercand sa il integrez cu aer conditionat si cu telefonul mobil.

Download

[Arhiva proiectului](#)

Bibliografie

- [Python serial](#)
- [Arduino serial](#)

- [ATmega328P Datasheet](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/avaduva/room-monitor>



Last update: **2021/06/04 18:05**