

# Home Monitoring

## Autor

[Bărbieru Andrei-Petre](#)

## Introducere

Proiectul consta in crearea unui server web accesibil printr-un browser, care monitorizeaza informatii despre camera in care este plasat si ofera acces si control asupra unui alt server care face stream la o camera video, amplasata in aceeasi incapere. De asemenea, o alarma poate sa fie activata din browser (de exemplu cand se vede ceva suspect pe camera), iar aceasta poate fi oprita doar prin introducerea unei parole folosind un keypad.

Scopul acestuia este supravegherea unei camere, iar ideea de la care am plecat a fost montarea unei camera video indreptate spre usa de la intrarea casei, pe care am dus-o putin mai departe.

Proiectul este util pentru mine deoarece pot urmari cu usurinta temperatura si calitatea aerului din camera in care se afla dispozitivul si sa urmaresc camera video live, toate acestea ori de pe telefon, ori de pe calculator.

## Descriere generală

### Schema bloc:



## Funcționalitate

Utilizatorul se conecteaza prin browser folosind adresa ip asignata. Acesta poate vedea informatiile de temperatura, umiditate si miscare in timp real (acestea se updateaza la cateva secunde). De asemenea, utilizatorul are acces la stream-ul video si are la dispozitie diferite butoane care controleaza camera, inclusiv un buton care face o captura la video si o downloadeaza automat.

De asemenea, utilizatorul are in browser un buton de alarma, care porneste un buzzer care poate sa

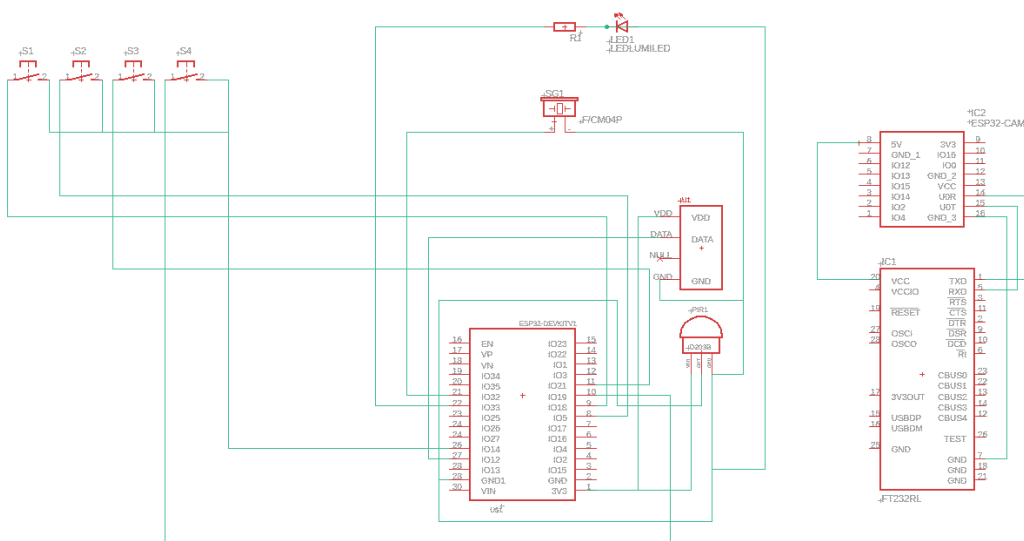
fie dezactivat doar in mod fizic, atunci cand se introduce parola corecta pe un keypad.

# Hardware Design

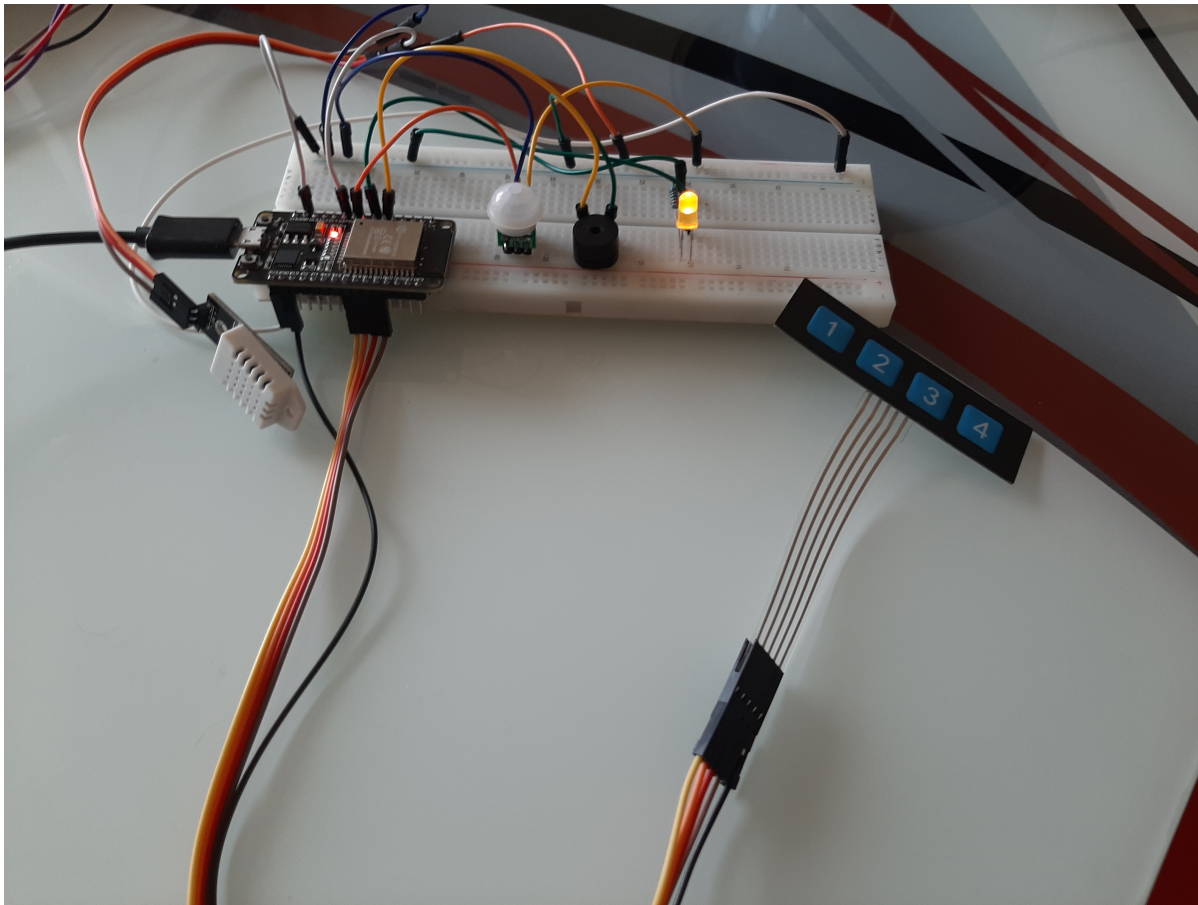
## Listă de piese

Produs	Link
Placa dezvoltare ESP32	<a href="https://www.emag.ro/placa-dezvoltare-esp32-devkit-v1-ai669-s332/pd/DXV9FDMBM/">https://www.emag.ro/placa-dezvoltare-esp32-devkit-v1-ai669-s332/pd/DXV9FDMBM/</a>
Modul ESP32-CAM	<a href="https://www.emag.ro/modul-esp32-cam-ov2640-camera-inclusa-ai065-s72/pd/DKDGT8BBM/">https://www.emag.ro/modul-esp32-cam-ov2640-camera-inclusa-ai065-s72/pd/DKDGT8BBM/</a>
Convertor mini USB Serial	<a href="https://www.emag.ro/modul-convertor-mini-usb-serial-ft232rl-rosu-cu-mufa-usb-mini-ai097-s188/pd/DBPRXGBBM/">https://www.emag.ro/modul-convertor-mini-usb-serial-ft232rl-rosu-cu-mufa-usb-mini-ai097-s188/pd/DBPRXGBBM/</a>
Senzor temperatura si umiditate	<a href="https://www.emag.ro/senzor-umiditate-si-temperatura-dht22-cl170/pd/DD006JBBM/">https://www.emag.ro/senzor-umiditate-si-temperatura-dht22-cl170/pd/DD006JBBM/</a>
Senzor miscare	<a href="https://www.emag.ro/senzor-miscare-am312-mini-arduino-ai003-s10/pd/DYV2JZBBM/">https://www.emag.ro/senzor-miscare-am312-mini-arduino-ai003-s10/pd/DYV2JZBBM/</a>
Fire mama-mama	<a href="https://www.emag.ro/modul-esp32-cam-ov2640-camera-inclusa-ai065-s72/pd/DKDGT8BBM/">https://www.emag.ro/modul-esp32-cam-ov2640-camera-inclusa-ai065-s72/pd/DKDGT8BBM/</a>
Fire mama-tata	<a href="https://www.emag.ro/set-10-fire-dupont-mama-tata-30-cm-cl95/pd/D4J66JBBM/">https://www.emag.ro/set-10-fire-dupont-mama-tata-30-cm-cl95/pd/D4J66JBBM/</a>
Fire tata-tata	
Breadboard 830p	
Keypad	<a href="https://ardushop.ro/ro/home/1003-tastatura-cu-4-intrari.html">https://ardushop.ro/ro/home/1003-tastatura-cu-4-intrari.html</a>

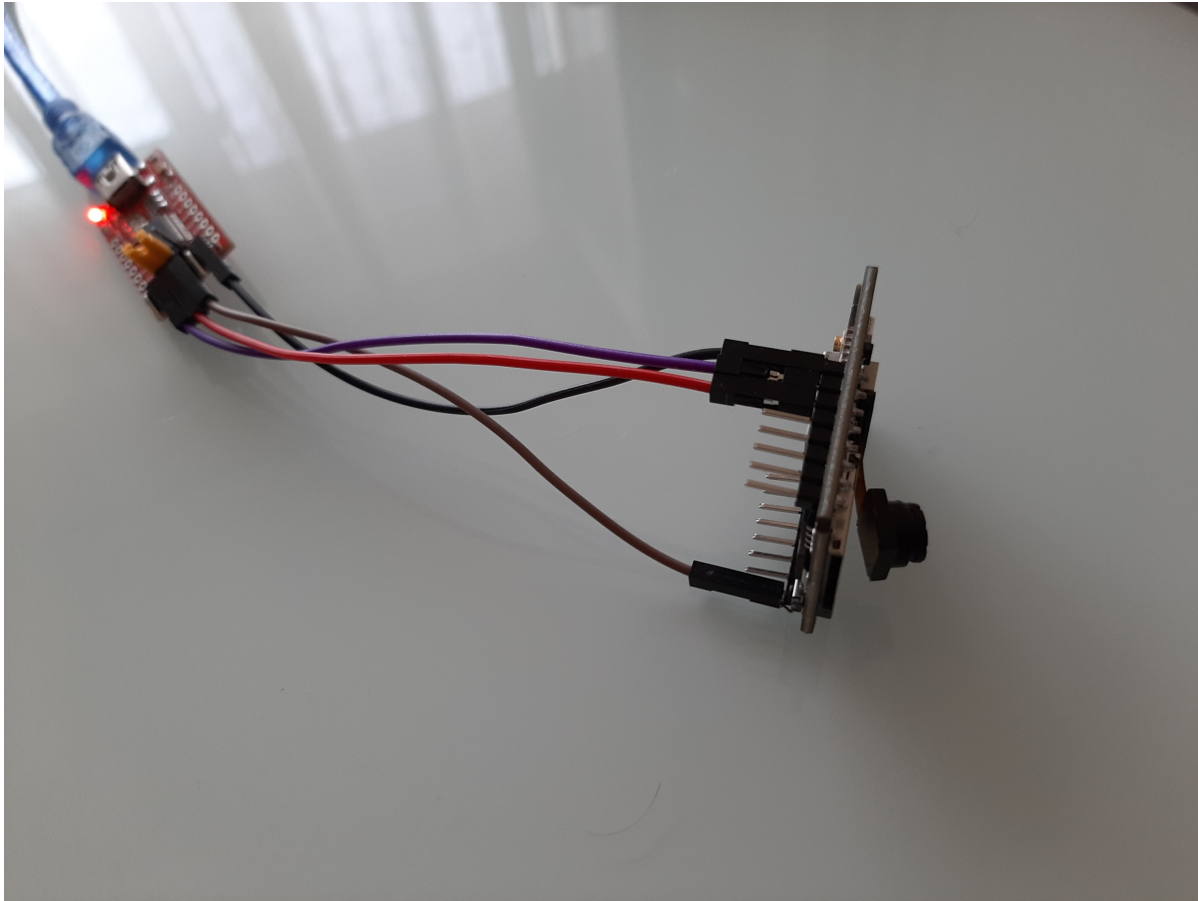
## Schema electrica



# Poze



Camera



## Software Design

## Mediu de Dezvoltare

Visual Studio Code cu extensia Arduino (Visual Studio Code extension for Arduino by Microsoft).

## Biblioteci

WiFi.h - folosit pentru conectarea la WiFi.

[ESPAsyncWebServer.h](#) - folosit pentru realizarea server-ului asincron si pentru template processing-ul oferit.

Datorita unui bug existent in momentul de fata (biblioteca foloseste simbolul % pentru procesarea variabilelor, de exemplu %variabila%, dar are probleme cand se gaseste simbolul % in css, de ex height:100%), am schimbat TEMPLATE\_PLACEHOLDER din '%' in '\$'. Am gasit un issue deschis pe github care vorbeste despre aceeaasi problema:

<https://github.com/me-no-dev/ESPAsyncWebServer/issues/981>

[DHT-sensor-library](#) - Folosit pentru citirea valorilor de temperatura si umiditate. Aceasta are ca dependenta si [Adafruit Unified Sensor Driver](#)

## Descriere implementare

In functia `setup()` initial se defineste rolul asignat fiecarui pin, dupa care se incearca conexiunea la WiFi pana cand aceasta reuseste. Dupa ce s-a conectat la WiFi, se scrie pe monitorul serial adresa ip locala a server-ului pentru a putea fi accesat din browser. De asemenea, tot aici se defineste raspunsul care trebuie returnat in urma unui request, pentru fiecare ruta care este folosita.

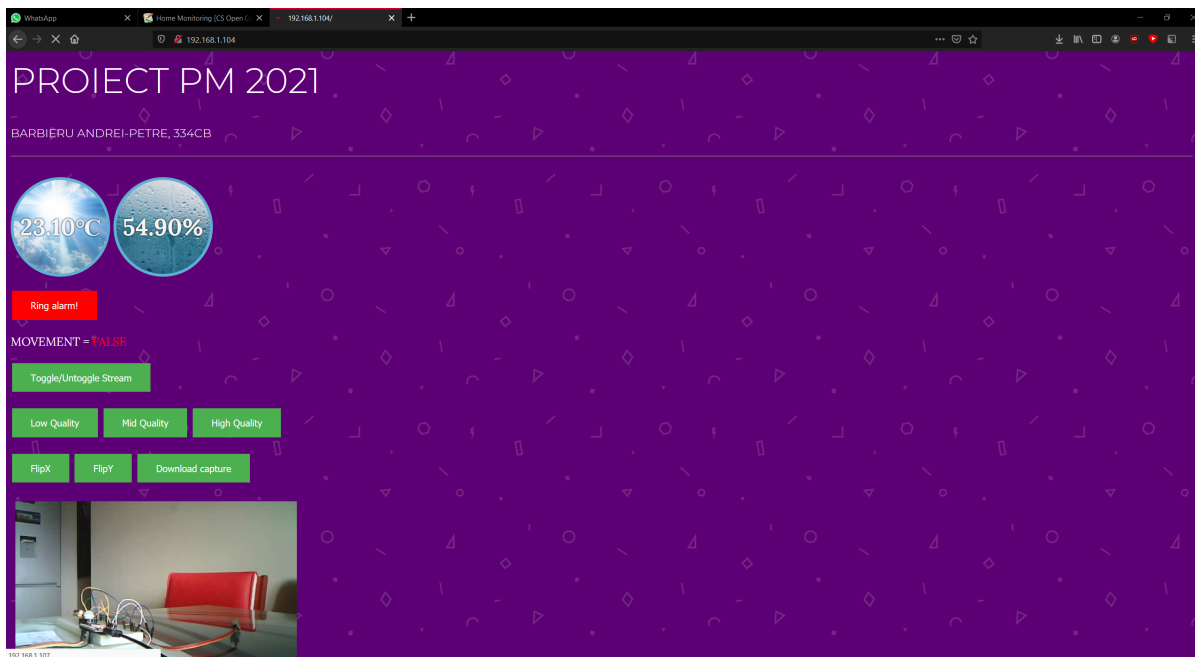
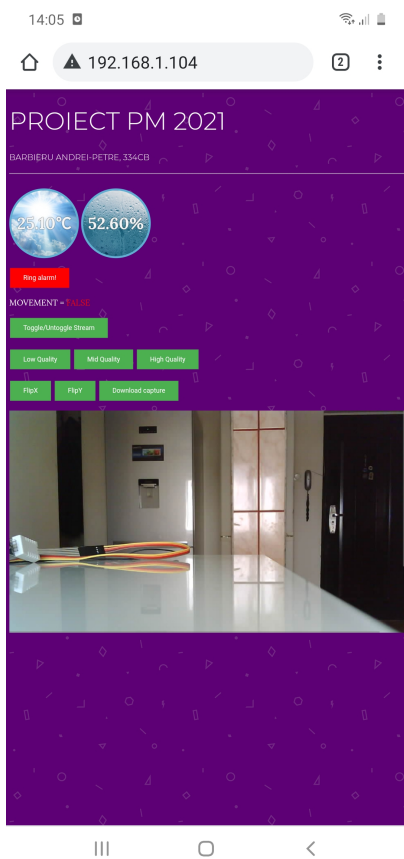
Am creat functia `getMotion()`, care citeste senzorul de miscare, si `readDHT(String val)`, care citeste senzorul de umiditate si temperatura pentru a le putea lua usor cand am nevoie de acestea.

Html-ul, css-ul si javascript-ul paginii este stocat in memoria flash, in PROGMEM (<https://www.arduino.cc/reference/en/language/variables/utilities/proGMEM/>). Aici se afla codul pentru styling-ul paginii, dar si pentru diferitele informatii oferite. Pentru updatarea valorilor pe site in timp real am creat cateva functii care se apeleaza la un anumit interval. In principiu, la un anumit numar de secunde se face un request de tip GET catre server-ul web care ii intoarce valorile cerute, in functie de calea request-ului. (am mentionat mai sus, sunt definite in `setup`).

Stream-ul live care se poate vedea pe pagina este hostat pe un esp32-cam. Codul pentru camera este cel oferit in `Examples→CameraWebServer`. Am urmarit codul si am adaugat butoane pentru controlul calitatii video si flip-ul imaginii video in ambele directii. De asemenea, am adaugat un buton care face captura la imaginea video oferita si o downloadeaza automat. Toate acestea sunt facute prin intermediul unor requesturi catre esp32-cam. Desi server-ul principal este asincron si permite mai multi utilizatori simultan, camera web este mai limitata si permite un singur utilizator sa fie conectat la un anumit moment dat. Asadar, desi pot accesa server-ul de monitorizare de pe mai multe device-uri simultan, se poate viziona stream-ul propriu-zis doar de pe un singur device (primul venit). Toate celelalte functionalitati sunt inca disponibile indiferent de numarul de utilizatori.

Intr-un final, mi s-a parut o idee buna sa pot activa o alarma din browser (de exemplu in caz ca se vede ceva suspect pe camera). Am creat un buton de alarma care face un request de tip get pe server. Acesta da drumul la buzzer care face zgomot pana cand aceasta este dezactivata. Singurul mod prin care alarma este dezactivata este prin introducerea parolei potrivite cu ajutorul unui keypad. Pentru realizarea acestui lucru, citesc mereu in loop valorile butoanelor de pe keypad. Retin mereu ultima valoare citita pentru a o lua in considerare o singura data (sa nu se puna aceeasi cifra de mai multe ori daca se tine apasat pe buton, etc). Deoarece butoanele de pe keypad sunt 1,2,3 si 4, am decis sa folosesc o parola de 5 cifre formata din acestea. De fiecare data cand se apasa pe un buton, se inmulteste o "parola curenta" cu 10 si se adauga numarul apasat. Daca parola curenta si parola reala au acelasi numar de cifre, se compara, altfel se trece mai departe. Indiferent daca parolele au fost identice sau nu, se reseteaza parola curenta cu 0. Cand parola curenta este 0, se tine aprins un led, care reprezinta faptul ca buffer-ul parolei este gol.

# Rezultate Obținute



[Playlist youtube](#)

## Concluzii

- Proiectul este realizat integral, implementand toate functionalitatile propuse.
- Permite utilizatorului sa vizioneze usor informatile vrute si sa urmareasca si controleze usor camera video.
- Pot sa fie adaugati usor alti senzori si sa fie integrati in program ca functionalitati suplimentare.

## Download

Codul .ino si schema EAGLE

[proiect\\_pm\\_home\\_monitoring.zip](#)

[Export to PDF](#)

## Jurnal

26 aprilie : alegerea temei proiectului

05 mai : ajunge prima comanda

19 mai : ajunge o comanda suplimentara

30 mai : finalizare wiki

luna mai: lucrat la proiect

## Bibliografie/Resurse

<https://randomnerdtutorials.com/program-upload-code-esp32-cam/>

[https://github.com/me-no-dev/ESPAsyncWebServer/blob/master/examples/simple\\_server/simple\\_server.ino](https://github.com/me-no-dev/ESPAsyncWebServer/blob/master/examples/simple_server/simple_server.ino)

<https://github.com/me-no-dev/ESPAsyncWebServer#send-large-webpage-from-program-containing-templates>

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/home-monitoring>



Last update: **2021/05/31 09:33**

