

# Stabilizator de camera selfie

**Autor:**

Moraru Teodora, 335CA

## Introducere

Proiectul consta intr-un stabilizator de camera selfie, care va putea fi controlat in doua moduri (manual sau automat), cu scopul de a centra camera pe fata utilizatorului. Poate fi utilizat in timpul unei conferinte online sau chiar pentru crearea mai facila a filmuletelor de vlogging.

## Descriere generală

Pentru controlul unghiului stabilizatorului, voi atasa camera de un servomotor. In modul manual, acesta va fi controlat cu ajutorul a doua butoane (left si right). In modul automat, va fi controlat in functie de pozitia fetei, detectate prin OpenCV. Detectia se va face in PC, iar comenzile pentru servomotor vor fi transmise catre Arduino prin seriala. Interschimbarea dintre cele doua moduri de utilizare se va face actionand un buton si va fi semnalata prin aprinderea unui led si pornirea unui buzzer.

## Hardware Design

In ceea ce priveste procurarea sau achizitionarea pieselor, am comandat un set complet pentru Arduino. Din acest set, am folosit doar piesele necesare acestui proiect. Set-ul poate fi gasit [aici](#).

**Lista componente:**

1. Arduino UNO
2. Breadboard
3. 3 x butoane
4. servomotor
5. 2 x rezistente
6. 2 x LED-uri
7. buzzer
8. fire

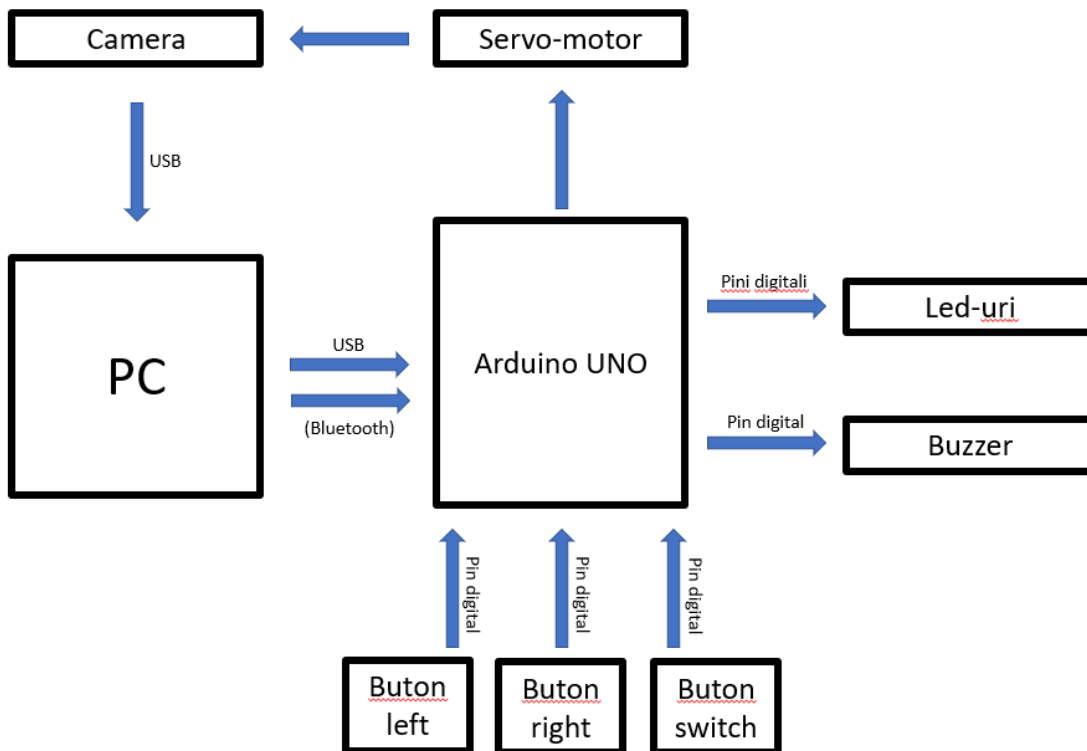
## 9. camera web

### Camera Web

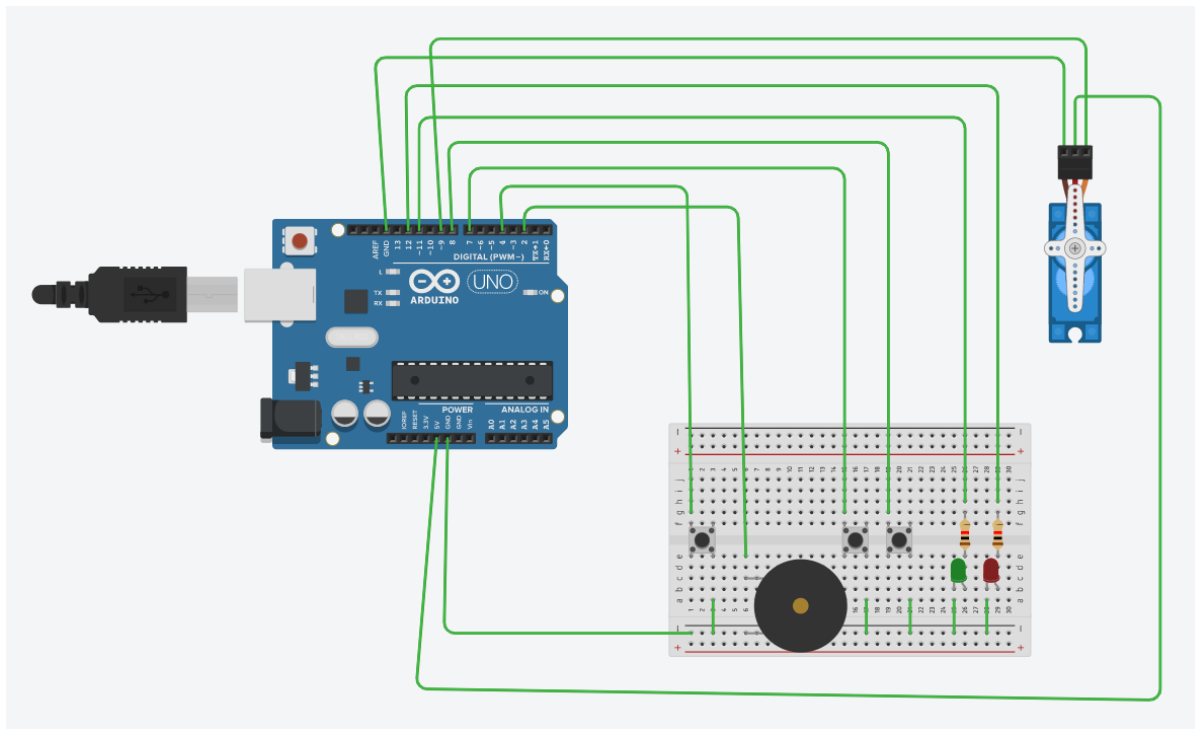


Camera web, SriHome™ SH039 Pro, Extra FullHD 3MP, unghi 110 grade, 30FPS, anulare zgomot de fond, rotire, plug & play, negru.  
Modelul poate fi gasit [aici](#).

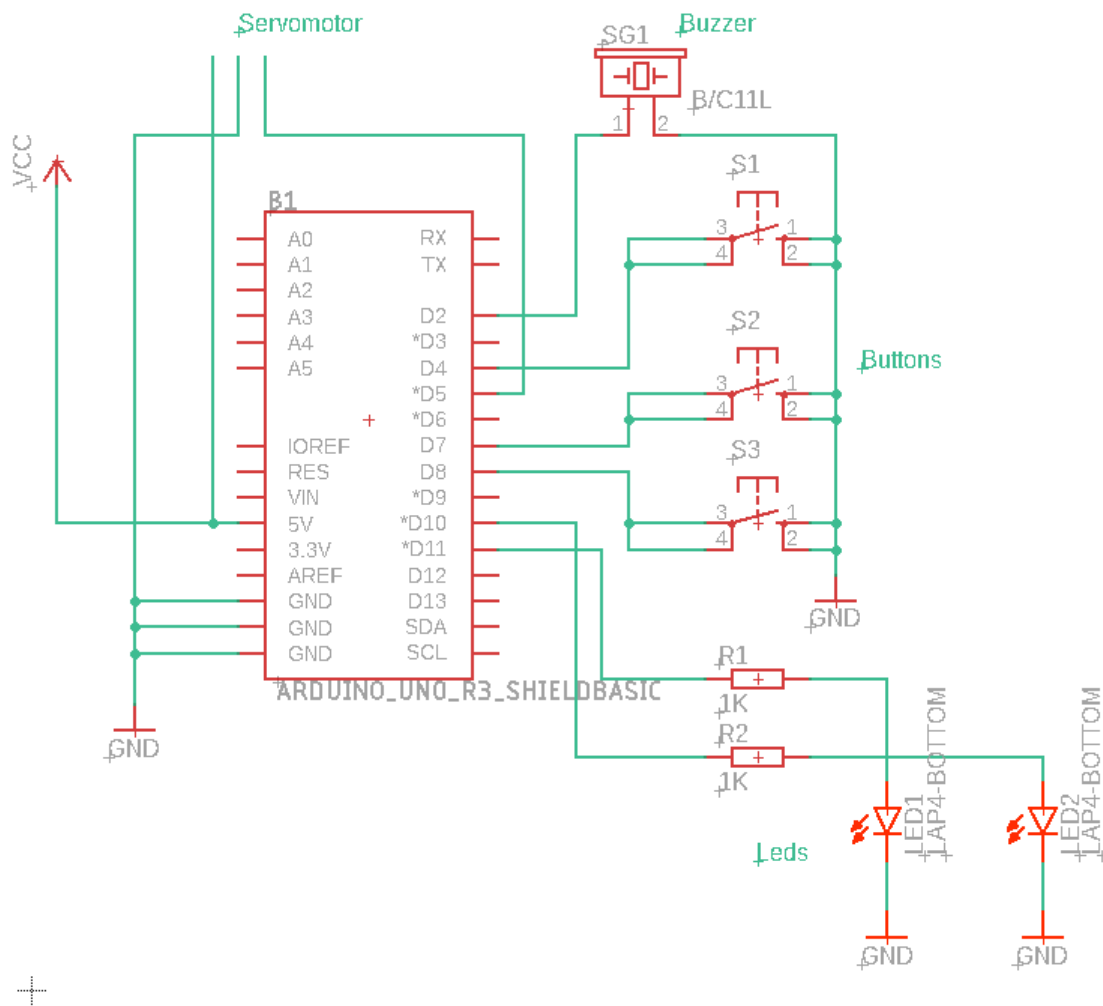
### Schema bloc:



### Schema Tinkercad:



Schema electrica:



# Software Design

## Continutul arhivei:

1. projectPM.ino
2. camera.py
3. haarcascade\_frontalface\_default.xml
4. README

Componenta software corespunzatoare placii Arduino e reprezentata de *projectPM.ino* si a fost scrisa in Arduino IDE. Pentru a putea lucra cu functiile necesare controlului servomotorului, am inclus biblioteca *Servo.h*. Algoritmul incepe cu initializarile parametrilor de stare si cei ai servomotorului, apoi continua cu cele doua functii: setup si loop. In functia **setup**, setez rolul pinilor de intrare si de iesire, starea initiala a celor doua LED-uri (verde aprins, rosu stins), atasez servomotorul pinului 9 de tip PWM si pornesc protocolul de comunicare prin seriala.

In functia **loop**, incep prin a face toggle pe butonul de switch: compar starea butonului cu cea anterioara si, daca aceasta difera, atunci se interschimba modurile de functionare ale stabilizatorului, respectiv se interschimba starile LED-urilor. Tot la apasarea butonului de switch, se porneste buzzer-ul, pentru a anunta auditiv utilizatorul ca s-a schimbat modul de utilizare. Salvez apoi starea curenta a butonului de switch pentru a fi folosita la urmatoarea iteratie. In cele ce urmeaza, algoritmul e impartit in doua cazuri:

1. **modul manual**: se citesc starile celor doua butoane (*left* si *right*). Daca unul dintre ele este apasat, se modifica gradul servomotorului in sensul dorit, fiind intotdeauna constrans de intervalul (-180, 180).
2. **modul automat**: se primeste prin seriala (de la algoritmul camera.py explicat mai jos) o valoare care este convertita la int. Aceasta valoare reprezinta un cod pentru unul dintre cele doua cazuri: necesitatea deplasarii la stanga sau la dreapta. In functie de caz, se seteaza gradul servomotorului, cu o granularitate de 5 grade.

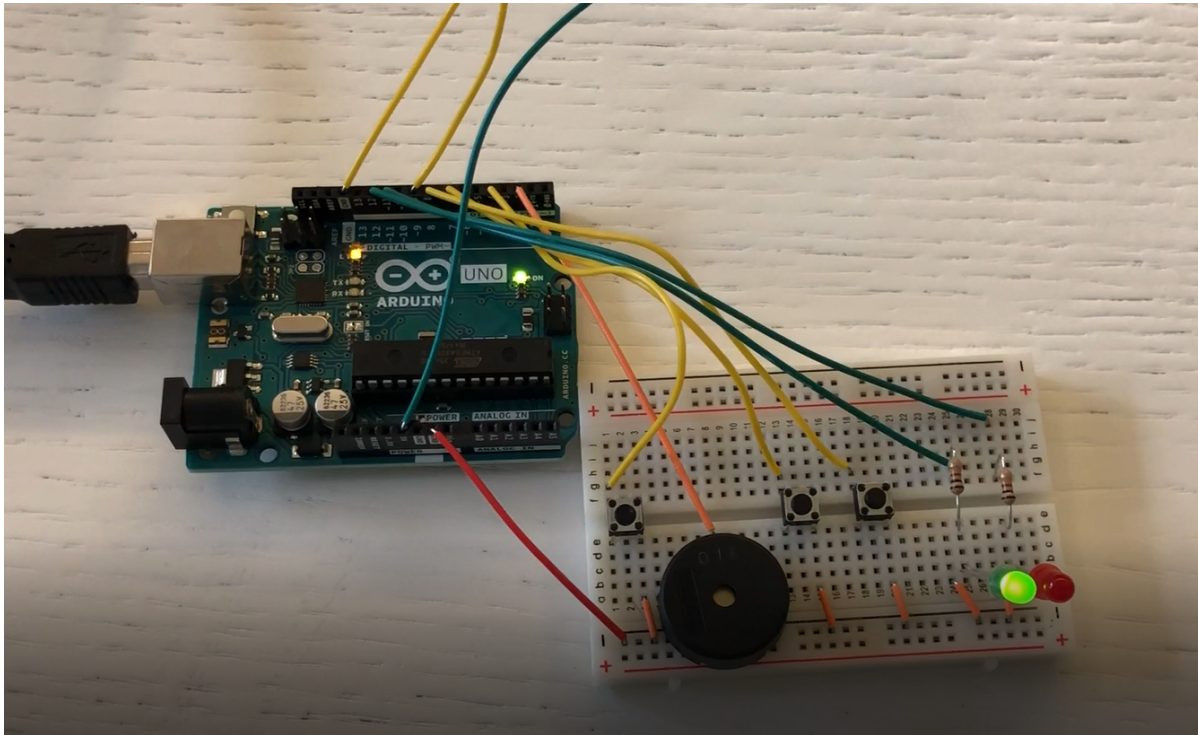
Pe de alta parte, *camera.py* (scris in PyCharm) reprezinta componenta software responsabila de pornirea camerei web, captarea si afisarea imaginilor, detectia fetei si transmiterea comenzilor corecte catre placa Arduino, prin protocolul de comunicare seriala. Pentru inceput, se seteaza portul si baudrate-ul pe care se va realiza comunicarea, cu ajutorul bibliotecii *serial*. Apoi, pentru a putea facilita detectia fetei, am incarcat o baza de date publica numita *haarcascade\_frontalface\_default.xml*. Captarea si afisarea imaginilor de la camera web se face cu ajutorul bibliotecii *cv2*.

Pasii de procesare sunt urmatarii: se capteaza imaginea, se salveaza coordonatele fetelor detectate, se itereaza prin ele pentru a gasi fata cea mai apropiata (metoda de eliminare a zgomotului), si apoi se deseneaza pe imagine un patrat cu acele coordonate. Se compara centrul patratului cu intervalul din mijloc de pe latimea imaginii, iar daca se afla in stanga sau in dreapta acestuia, se transmite prin seriala codul corespunzator deplasarii. Un aspect interesant este ca aceasta transmisie e asincrona, asa ca, pentru a ma asigura ca algoritmul continua abia dupa ce mesajul a fost primit la destinatie, astept un raspuns inapoi de la placa (aka *ack* - cu riscul microrarii framerate-ului interfetei catre utilizator). La finalul buclei, dau flush seriei pentru a ma asigura ca nu ramane nimic in buffer. Iesirea din interfata si inchiderea programului se face prin apasarea tastei *ESC*. Tot atunci se inchide camera si protocolul serial.

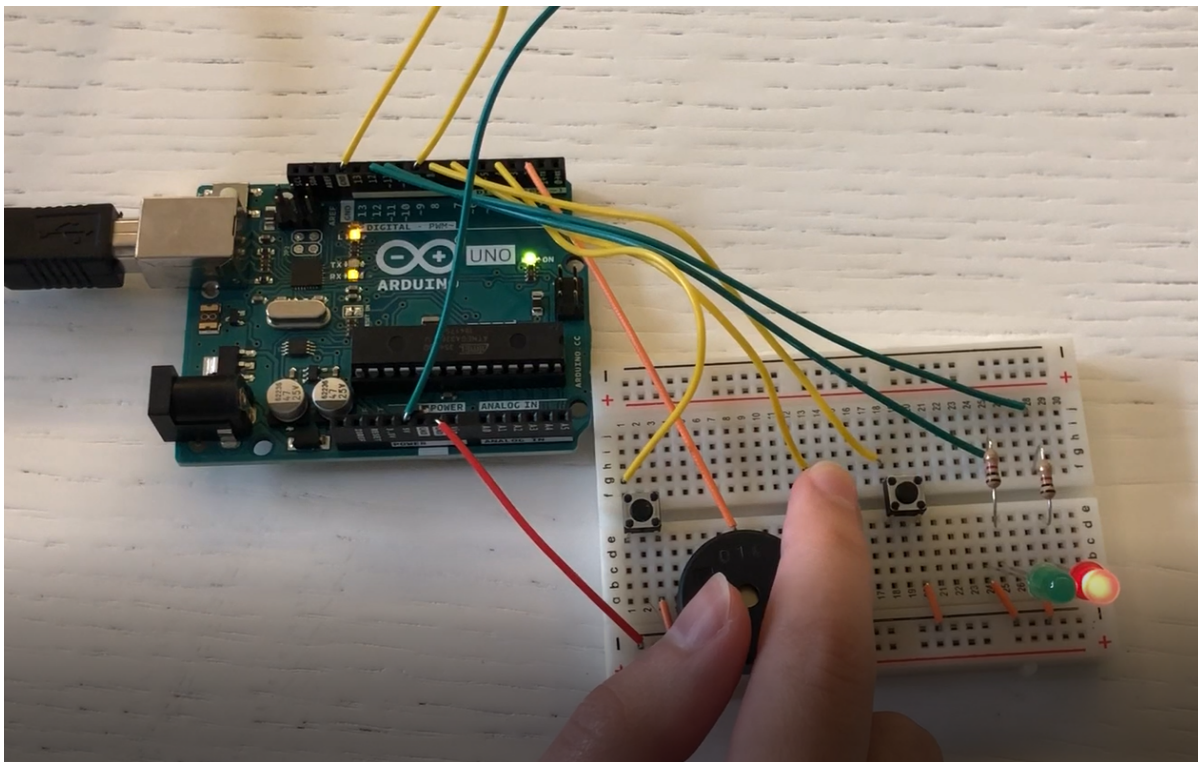
# Rezultate obtinute

Aici puteti viziona un [demo video](#).

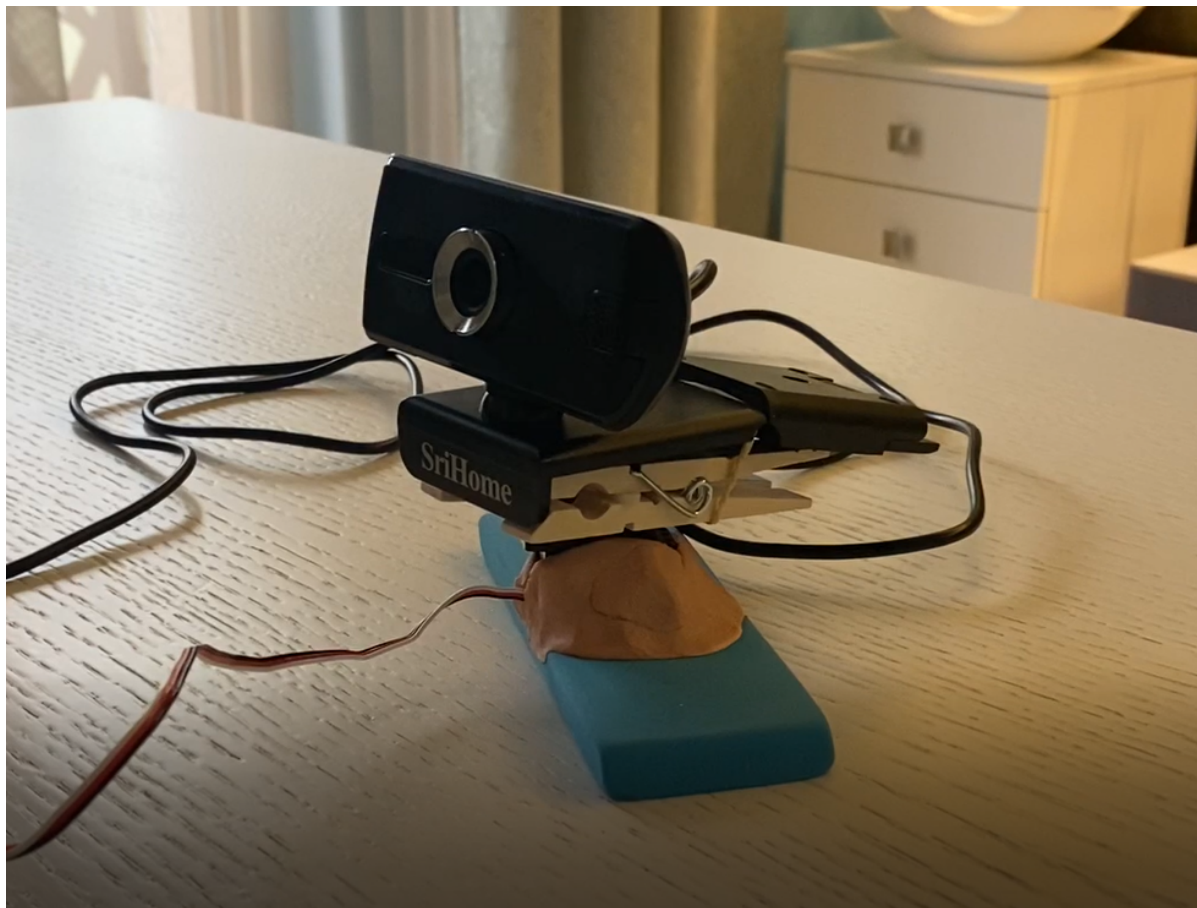
## Circuit in modul automat:



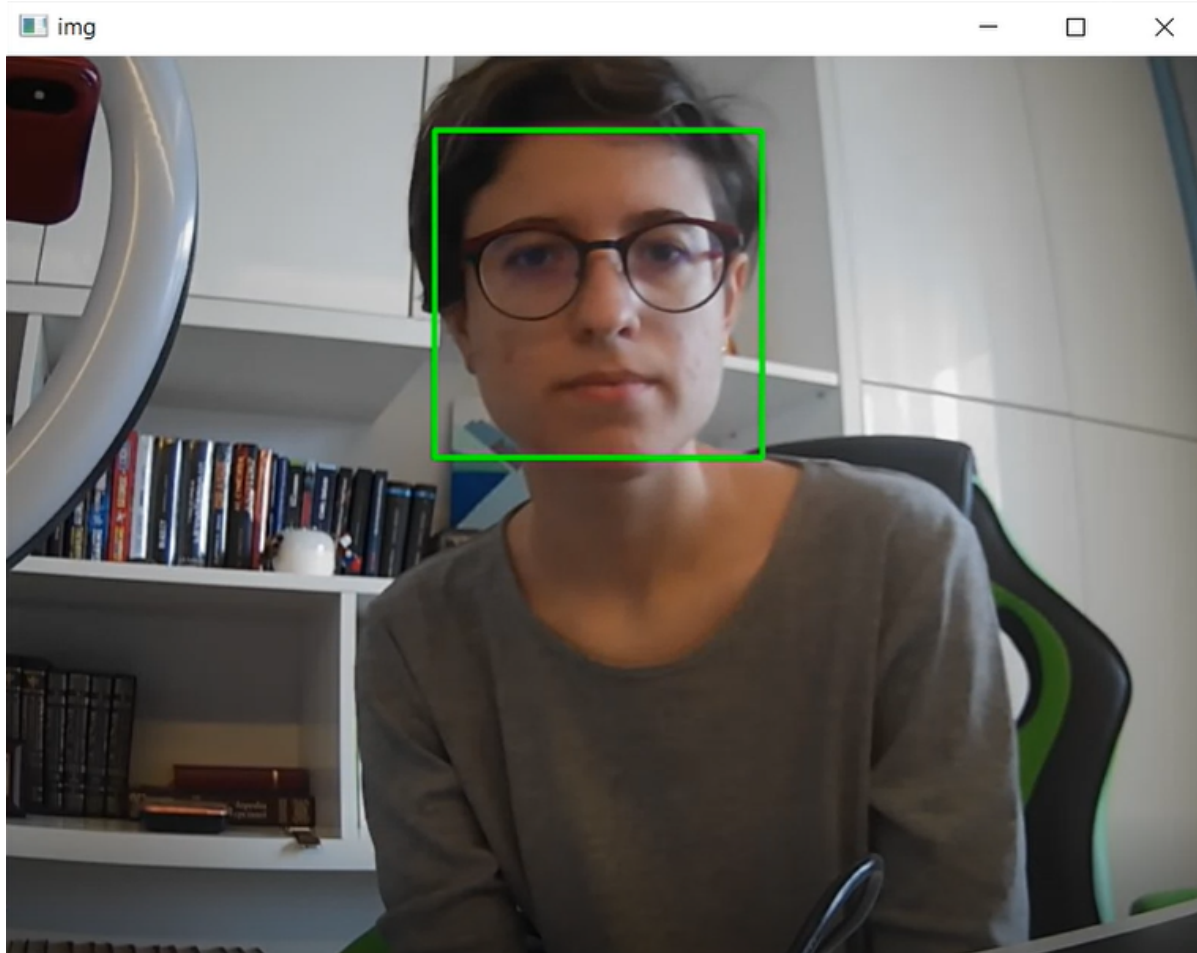
## Circuit in modul manual:



## Camera web conectata pe servomotor:



**Interfata utilizatorului:**



# Concluzii

Mi-a placut foarte mult sa lucrez la acest proiect, deoarece am avut sansa sa aprofundez cunostintele acumulate la curs si la laborator pentru a realiza un prototip interesant si foarte util. Sunt constienta ca functionalitatile sale nu sunt extrem de avansate, asa ca imi propun ca in viitorul apropiat sa dezvolt si mai mult acest prototip (sa creez o interfata grafica mai prietenoasa, sa mai adaug un servomotor pentru deplasarea verticala etc.)

# Download

De aici se poate descarca arhiva:

[stabilizator.zip](#)

# Jurnal

- 25.04.2021 - Alegerea temei si crearea paginii de wiki
- 03.05.2021 - Comandarea camerei web
- 16.05.2021 - Realizarea schemelor hardware
- 22.05.2021 - Finalizarea conectarii componentelor hardware
- 30.05.2021 - Finalizarea componentelor software
- 02.06.2021 - Finalizarea paginii de wiki

# Bibliografie/Resurse

[Laborator 0 PM](#)  
[Laborator 2 PM](#)  
[Laborator 3 PM](#)  
[Button Debounce Example](#)  
[Face Detection Example](#)  
[Serial Communication in Python Example](#)  
[Laborator EAGLE](#)

[Export to PDF](#)

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

Permanent link:  
[http://ocw.cs.pub.ro/courses/pm/prj2021/alazar/stabilizator\\_camera\\_selfie](http://ocw.cs.pub.ro/courses/pm/prj2021/alazar/stabilizator_camera_selfie)



Last update: **2021/06/02 17:07**