

RedSight: Tintă controlată prin recunoaștere facială

Introducere

Prezentarea pe scurt a proiectului vostru:

- ce face
- care este scopul lui
- care a fost ideea de la care ați pornit
- de ce credeți că este util pentru alții și pentru voi

Proiectul consta într-o tinta automatizata care foloseste un modul ESP32-CAM pentru a detecta si urmari in timp real fata unei persoane. Atunci cand camera recunoaste o fata prestabilita, isi ajusteaza tinta si o urmareste cu ajutorul unui laser rosu.

- Ce face: detecteaza fete, le urmareste automat cu ajutorul unui laser.
- Scopul: crearea unui sistem interactiv cu detectare faciala
- Ideea de la care am pornit: mi-am dorit sa explorez capabilitatile ESP32-CAM in procesare de imagini si AI
- De ce este util: proiectul demonstreaza cum pot fi combinate detectarea faciala si miscarea controlata într-un sistem embedded, fiind relevant pentru aplicatii precum supraveghere si securitate

Descriere generală

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează. Exemplu de schemă bloc:

<http://www.robs-projects.com/mp3proj/newplayer.html>

Schema BLOCK a proiectului



Proiectul este alcatuit din urmatoarele componente:

- **ESP32-CAM** - microcontroler cu Wi-Fi, Bluetooth si camera integrata, folosit ca unitate principala de control
- **Ecran LCD 1602 I2C** - afisaj cu doua randuri, conectare usoara prin I2C, folosit pentru afisarea informatiilor accuracy-ului si a label-ului detectat de modelul de recunoastere faciale
- **Micro Servo SG90** - servomotor utilizat pentru a urmari o fata prestabilita cu o precizie intre 0° si 180°
- **Dioda laser 5V** - emite un fascicul laser, util pentru tintire
- **Modul joystick doua axe XY** - senzor analogic pentru a putea controla tinta manual

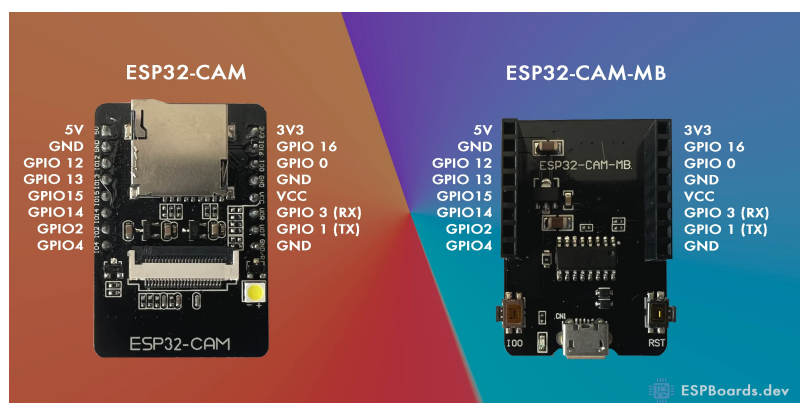
Hardware Design

Aici puneți tot ce ține de hardware design: listă de piese scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>) diagrame de semnal rezultatele simulării

Lista de piese

Componenta	Descriere	Link	Datasheet
ESP32-CAM	Modul ESP32 cu camera OV2640	ESP32-CAM	Datasheet
Micro servo SG90	Servomotor 180°	SG90	Datasheet
Dioda Laser 5V	Laser pentru tintire	Laser	-
Ecran LCD 1602 I2C	Ecran LCD pentru afisare	LCD 1602	Datasheet
Modul Joystick XY	Modul pentru a controla tinta manual	Joystick XY	Datasheet

Pini utilizati



Componentă	Pini	Interfață
ESP-CAM	GND, 5V, GPIO 4, GPIO 14, GPIO 13, RX, TX	-
ESP-CAM-MB	GND, 5V, RX, TX	UART
Laser	GND, 5V	Digital OUT
Servomotor SG90	GND, 5V, GPIO 4	PWM
Ecran LCD	GND, VCC, SDA, SCL	I2C

Schema Electrica

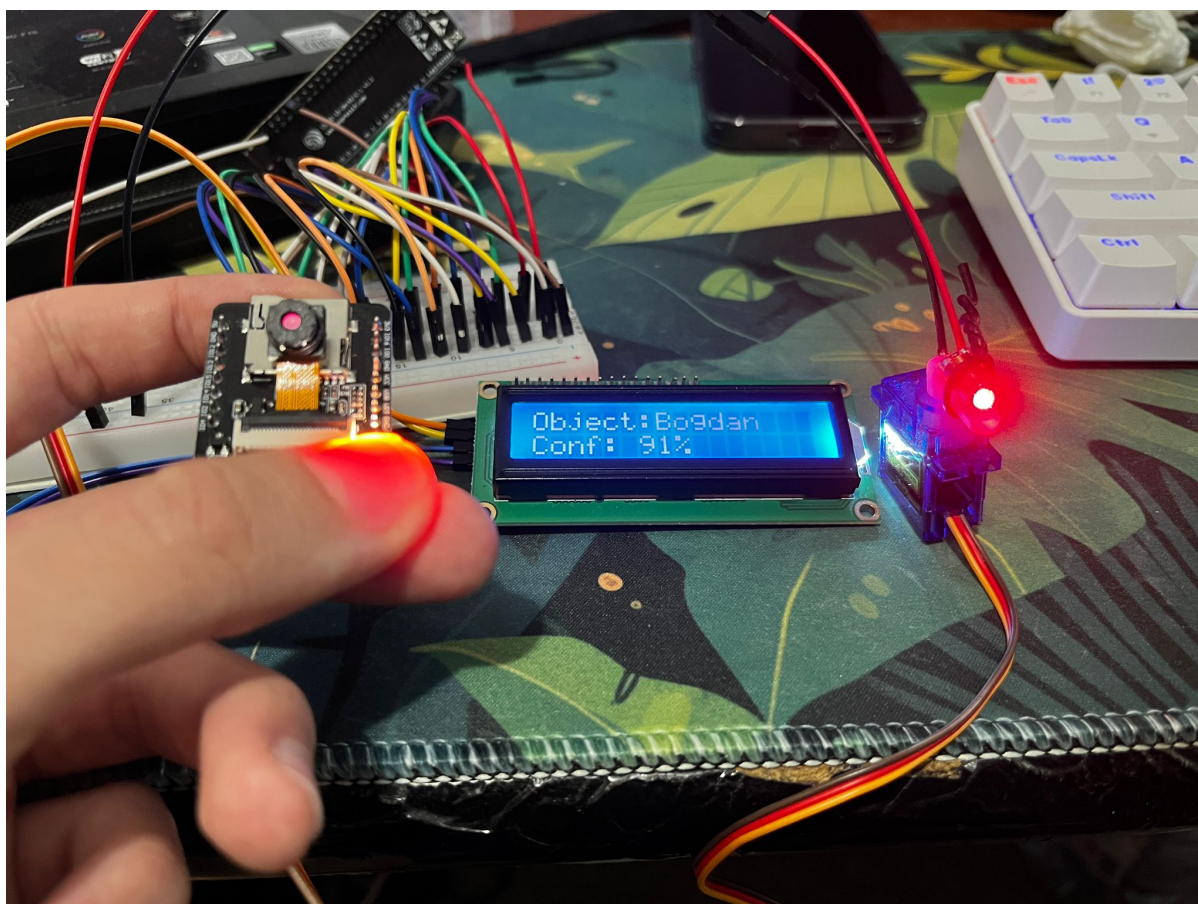


Calculul consum energie

Componenta	Consum estimat
ESP32-CAM	~160 mA (Wi-Fi activ)
Micro servo SG90	~100-250 mA (in sarcina)
Dioda Laser 5V	~30 mA
LCD 1602 I2C	~20 mA
Modul Joystick XY	~5 mA

Rezultate

Toate componentele au fost testate in simulare si functioneaza corespunzator, interactionand corect cu ESP32-CAM. Sistemul raspunde eficient atat la comenzile automate (detectarea fetei), cat si la cele manuale (prin joystick). Afisajul LCD reda in timp real informatiile esentiale, iar laserul si servomotorul colaboreaza pentru urmarirea si indicarea precisa a tintei.



Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuieți să le implementați
- (etapa 3) surse și funcții implementate

Mediu de dezvoltare

Aplicația firmware este dezvoltată în Arduino IDE, utilizând extensia corespunzătoare pentru plăcile ESP32 și suport pentru camera AI Thinker (OV2640). Mediul oferă integrare cu portul serial și suport pentru biblioteci externe utilizate pentru inferență ML, control servomotor și afișare LCD.

Librării și surse 3rd-party utilizate:

- **Edge Impulse SDK:** utilizat pentru rularea inferenței de recunoaștere facială și procesarea imaginii capturate.
- **ESP32-Camera Library:** permite inițializarea și capturarea de cadre cu camera OV2640.
- **ESP32Servo:** permite controlul unui servomotor PWM conectat la ESP32.
- **LiquidCrystal_I2C:** pentru afișarea textelor pe un ecran LCD 16×2, comunicând prin protocol I2C.
- **Wire.h:** utilizată pentru comunicarea I2C cu LCD-ul.

Algoritmi și structuri implementate/plănuite:

- **Captură și procesare imagine:** imaginea este capturată la rezoluție HD (1280×720) în format grayscale, convertită în RGB888 și redimensionată pentru modelul ML.
- **Model ML de recunoaștere facială (Edge Impulse):** modelul a fost antrenat în Edge Impulse Studio pentru a detecta și clasifica fețe. Setul de date include imagini ale unei persoane-țintă („Bogdan”), precum și alte clase pentru generalizare.
- Detalii despre model:
 - **Tip:** Object Detection
 - **Rezoluție:** 96×96
 - **Format imagine:** GRAYSCALE
 - **Număr de epoci:** 60
 - **Algoritm:** FOMO (Faster Objects, More Objects) MobileNetV2 0.35
- Parametri:
 - **Features:** 2 (Bogdan, no_Bogdan)
 - **Learning rate:** 0.01
 - **Data Augmentation:** activat

- **Split date:** 80% train, 20% validation
- **Batch size:** 32
- **Input layer:** 9,216 features
- **Control servo:** dacă este detectată o față etichetată „Bogdan” cu încredere > 0.5 , se calculează centrul bounding box-ului și se rotește servomotorul pentru a “urmări” această față.
- **Afișare pe LCD:** este afișată în timp real eticheta obiectului cu cea mai mare încredere și procentul acestei încrederi.
- **Gestionare memorie:** bufferul de imagine (snapshot_buf) este alocat dinamic și eliberat după fiecare buclă de inferență.
- **Inferență continuă:** funcția loop() captează cadre, rulează inferența și actualizează starea servo și LCD într-o buclă infinită.

Face recognition - ESP32-CAM & Edge Impulse

Platforma folosita

- Modelul de recunoastere faciala a fost realizat pe **Edge Impulse Studio**.

Date utilizate pentru antrenare si testare

- Sursa imaginilor: poze realizate atat cu **ESP32-CAM**, cat si cu **telefonul mobil**.
- Numar de imagini in setul de antrenare: **365**
- Numar de imagini in setul de test: **88**

Modelul antrenat

- Versiune model: **Quantized**
- Numar cicluri de antrenare: **60**
- Optimizator invatat: **Activat**
- Learning rate: **0.01**
- Procesor de antrenare: **CPU**
- Augmentare date: **Activata**
- Batch size: **32**
- Dimensiune set de validare: **20%**

Performante pe setul de validare

Confusion Matrix:



Metrici generale:

- **Precision (non-background):** 0.84
- **Recall (non-background):** 0.88
- **F1 Score (non-background):** 0.86
- **Accuracy:** 75.00%

Performanta On-Device

Procesare imagini:

Etapa	Timp	RAM
Pre-procesare	15 ms	4 KB
Inferenta	1122 ms	119.4 KB
Total	1137 ms	239.4 KB

Stocare:

- **Flash usage (total):** 90.9 KB

Rezumat performanta

Metriaca	Valoare
Timp total	1137 ms
RAM totala	239.4 KB
Flash total	90.9 KB
Acoperire F1	88.3%

Concluzii

Modelul de recunoastere faciala dezvoltat pe platforma Edge Impulse Studio pentru ESP32-CAM ofera o performanta buna atat in ceea ce priveste acuratetea, cat si eficienta pe dispozitiv. Cu un scor F1 de 89.7% pe setul de validare si o precizie de 93.8% pentru clasele non-background, modelul este capabil sa diferentieze corect intre persoanele recunoscute si fundal, chiar si in conditii variate, folosind imagini capturate atat cu ESP32-CAM, cat si cu telefon mobil.

Timpul de procesare de aproximativ 1.1 secunde per inferenta este rezonabil pentru aplicatii embedded, iar consumul redus de memorie RAM (sub 240 KB) si spatiu in flash il fac potrivit pentru implementarea pe microcontrolere cu resurse limitate, precum ESP32-CAM.

Surse și funcții implementate

- `setup()`: Inițializează componentele hardware: portul serial, camera, LCD-ul, servomotorul și pornește o pauză inițială de 2 secunde.
- `loop()`: Se ocupă de:
 - captură imagine și conversie,
 - rularea inferenței ML cu Edge Impulse,
 - detecția feței „Bogdan”,
 - controlul servomotorului pe baza poziției bounding box-ului,
 - actualizarea LCD-ului.
- `ei_camera_init()`: Configurează și inițializează camera AI Thinker (OV2640) cu pinii specifici plăcii ESP32-S3.
- `ei_camera_deinit()`: Dezactivează camera.
- `ei_camera_capture()`: Capturează cadrul curent, face conversia în RGB și redimensionează imaginea la dimensiunea cerută de modelul de inferență.

Pipeline



Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

În urma realizării proiectului, au fost obținute următoarele rezultate semnificative:

- A fost construit un sistem functional de recunoastere faciala folosind ESP32-CAM, capabil sa identifice in timp real persoana tinta
- Modelul de inteligenta artificiala antrenat pe platforma Edge Impulse Studio a atins o acuratete ridicata, cu un scor F1 de 88.3% si o precizie de 92% pe setul de validare.
- Sistemul ruleaza direct pe dispozitiv, avand un timp de inferenta de aproximativ 1.1 secunde, ceea ce il face potrivit pentru aplicatii embedded cu resurse limitate.
- Interfatarea componentelor externe precum servomotorul si ecranul LCD a permis extinderea functionalitatilor sistemului (e.g., afisarea de mesaje privind clasa dominanta din frame-ul respectiv).
- S-a realizat cu succes colectarea si prelucrarea de date atat de pe ESP32-CAM, cat si din surse externe (telefon mobil), contribuind la imbunatatirea modelului.
- Aceste rezultate demonstreaza viabilitatea proiectului si deschid oportunitati pentru dezvoltari ulterioare in domeniul securitatii si al interactiunii om-dispozitiv.

Concluzii

- Proiectul a demonstrat cu succes capabilitatile modulului **ESP32-CAM** in captarea si transmiterea imaginilor in timp real.
- Configurarea si programarea modulului au oferit o buna oportunitate de aprofundare a cunostintelor despre microcontrolere si comunicatii IoT.

Download

O arhivă (sau mai multe dacă este cazul) cu fişierele obţinute în urma realizării proiectului: surse, scheme, etc. Un fişier README, un ChangeLog, un script de compilare şi copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Edge Impulse Model Build:

- [ei-face-recognition.zip](#)

Cod proiect:

- [esp32_camera.zip](#)

Link Github proiect:

- [Github](#)

Demo YouTube:

- [Demo](#)

Pentru ca vor mai aparea modificari la cod, am facut si un repository de github unde voi uploada mereu varianta buna atunci cand apare vreo modificare.

Jurnal

Puteţi avea şi o secţiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

03.05.2025

Am cumparat toate piesele pentru proiect, inclusiv un esp32-s3-devkit1 pe care planuiesc sa il

folosesc.

12.05.2025

Ma chinui sa conectez camera externa la esp32-s3-devkit1, camera pe care o comandasem nu avea si conectorul inclus, a trebuit sa comand alta si sa astept dupa ea ceva timp.

14.05.2025

Am reusit sa conectez camera externa ov2640 la esp32-s3-devkit1, utilizand esp-idf. Mi-am creat si modelul de face recognition, incep usor usor sa il antrenez cu poze. Imi dau comanda si de un esp32-cam in caz de backup, cu tot cu programator.

18.05.2025

Din pacate, nu reusesc sa urc modelul pe esp32-s3-devkit1, asa ca ma mut cu proiectul pe esp32-cam.

20.05.2025

Reusesc sa urc modelul si sa implementez majoritatea functionalitatilor pe care le va avea proiectul meu pe esp32-cam.

25.05.2025

Toate componentele merg ok, mai sunt cateva schimbari minore de facut la cod si poate si ceva fine-tuning la model.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/bogdan.prisacaru>



Last update: **2025/05/28 13:33**