

# Sistem Automat de Țintire

**Nume: Antonescu Albert**  
**Grupa: 333CD**  
**Îndrumător: Ionuț Oțelea**

## Introducere

### Descriere

**Sistemul Automat de Țintire** este un **Auto-Turret** construit pe platforma **Arduino UNO**, care urmărește o țintă vizuală identificată cu ajutorul unui algoritm **OpenCV** rulat pe un calculator. Acesta simulează un **mecanism defensiv automat**, capabil să orienteze un dispozitiv pe două axe (orizontal și vertical) și să acționeze un trăgaci mecanic în momentul în care ținta este centrată.

Detecția se face cu o *cameră web* și recunoaștere de *cercuri roșii* (țintă) și *punct laser* (vizor), iar comenzile sunt transmise prin **UART** către microcontroler. Sistemul folosește **3 servomotoare**: două pentru mișcare și unul pentru trăgaci. Feedback-ul de stare este oferit printr-un **LED RGB** și un **buzzer**, pentru semnalizare vizuală și sonoră în timp real. Există și un **mod MANUAL**, controlat cu ajutorul unui **joystick analogic**, pentru testare și calibrare.

### Scopul și ideea principală

Scopul acestui proiect este de a realiza un **sistem embedded interactiv**, care să combine:

- controlul fizic al componentelor (servo, LED, buzzer),
- comunicarea serială între calculator și microcontroler,
- procesarea imaginii în timp real pentru decizie automată.

Proiectul a fost gândit ca o aplicație funcțională, cu o componentă vizuală spectaculoasă, dar și cu valoare educativă ridicată, ideală pentru aprofundarea noțiunilor din laboratorul de **Proiectarea cu Microprocesoare** (PM). Auto-Turret-ul ilustrează perfect o situație practică de integrare a mai multor concepte fundamentale din electronica digitală și programare bare-metal.

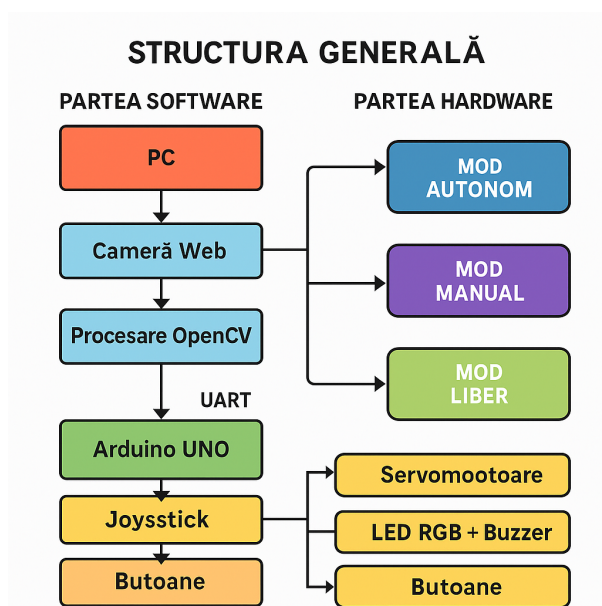
### Utilitatea

Proiectul este **modular, educativ** și ușor de adaptat pentru alte scopuri. El oferă un cadru concret pentru exersarea unor concepte importante precum **PWM, ADC, UART**, control pe **interuperi** și

manipulare **bare-metal a registrelor**.

Combinând partea de hardware cu un algoritm vizual extern, sistemul devine o platformă flexibilă pentru extinderi: poate fi dotat cu afișaj LCD, senzori de mișcare, module wireless sau chiar inteligență artificială. LED-ul RGB, buzzer-ul și joystick-ul adaugă o interfață intuitivă pentru utilizator, iar comenzile seriale permit integrarea ușoară cu orice aplicație de PC. Sistemul poate fi transformat ușor într-un **modul autonom** sau într-un **robot interactiv**.

## Structura generală



**Sistemul Automat de Țintire** este împărțit în două componente principale:

- **Partea software** - PC (Python + OpenCV)
- **Partea hardware** - Arduino UNO + componente electronice

Comunicarea între cele două se face prin **interfață UART** (USB ↔ Serial), prin care calculatorul transmite comenzi text către microcontroler. către microcontroler.

## Module software (PC)

### 1. Cameră Web (input video)

Captează imaginea în timp real și o trimite spre procesare.

### 2. Procesare OpenCV (Python)

- Detectează **cercuri roșii** (ținte) și **punctul laser** (vizor)
- Compară pozițiile pentru a determina dacă ținta este aliniată
- În funcție de poziții, trimite comenzi seriale către Arduino:

**PANU=..., TILTU=..., FIRE=1, MODE=...** etc.

## Module hardware (Arduino UNO)

### 1. Control servomotoare (PWM)

- Servo Pan (D9): control stânga-dreapta
- Servo Tilt (D10): control sus-jos
- Servo Fire (D11): acționează trăgaciul

### 2. LED RGB + Buzzer (feedback stare)

- Indică vizual și sonor starea sistemului:
  - Alb = liber (**FREE**)
  - Verde = scanare (**SCAN**)
  - Galben = țintă găsită (**LOCK**)
  - Roșu = avertizare (**FIRE\_WARN**)
  - Albastru = control manual (**MANUAL**)

### 3. Joystick (mod MANUAL)

- Control manual pentru Pan și Tilt (A4, A5)
- Buton pentru tragere (D13)

### 4. Butoane fizice (D5, D6, D7)

- Permite schimbarea modurilor prin apăsare:
  - D5 = **FREE**
  - D6 = **SCAN**
  - D7 = **MANUAL**

## Hardware Design

### Listă componente utilizate:

**Microcontroler:** Arduino UNO R3

**Servomotoare:** 3× SG90 (Pan, Tilt) + MG996R (trăgaci)

**Joystick:** modul PS2 (A4 - X, A5 - Y, D13 - buton)

**LED RGB:** cu 3 pini (conectat la D2, D3, D4)

**Buzzer pasiv:** conectat la D12

**Buton moduri:** 3× push-button pe D5, D6, D7 (cu rezistență pull-up)

**Camera Web:** conectată la PC (pentru OpenCV)

**Alte piese:** Breadboard, fire jumper, rezistențe 10kΩ, sursă externă 5V/2A

### Schemată electrică generală



### Diagrame semnal:

- Semnal PWM generat de funcția `servo.writeMicroseconds()` pentru Pan/Tilt/Fire.
- Semnal digital ON/OFF pentru LED-uri și buzzer.
- Citire ADC de la joystick (valorile analogice pe A4 și A5).

### Simulare și testare

- Fiecare componentă a fost testată individual înainte de integrare.
- Comenzile UART au fost validate cu Serial Monitor înainte de rularea OpenCV.
- S-au utilizat **delay-uri controlate** pentru sincronizarea trăgaciului și LED/buzzer.

### Simulare și testare

- Fiecare componentă a fost testată individual înainte de integrare.
- Comenzile UART au fost validate cu Serial Monitor înainte de rularea OpenCV.
- S-au utilizat **delay-uri controlate** pentru sincronizarea trăgaciului și LED/buzzer.

## Software Design

### Mediu de dezvoltare

- **PlatformIO + Visual Studio Code** - pentru codul C++ pe Arduino UNO.
- **Python 3.10** - pentru scriptul de detecție și control al sistemului.
- Testare interactivă cu **Serial Monitor** și **fereastră OpenCV**.

### Librării și surse externe

- **Servo.h** - pentru controlul servomotoarelor.
- **OpenCV** - pentru procesare imagine și detecție de forme.
- **PySerial** - pentru comunicare UART între PC și Arduino.
- **Numpy** - pentru calcule vectoriale și distanțe.

## Structura codului

### Pe Arduino (firmware)

- Implementare completă cu **scriere bare-metal** pentru: `digitalRead`, `digitalWrite`, `analogRead`.
- Control **PWM** pentru cele 3 servomotoare: Pan, Tilt, Fire.
- Control LED RGB + buzzer pentru feedback în timp real.

- Trimitere și recepționare comenzi prin **UART Serial**.
- Implementare logica modurilor: `FREE`, `SCAN`, `LOCK`, `FIRE\_WARN`, `MANUAL`.

## Pe PC (Python + OpenCV)

- Inițializare cameră și captură video în timp real.
- Detectie țintă: **HoughCircles** pentru identificarea unui cerc roșu (țintă).
- Detectie punct laser: bazat pe intensitatea canalului roșu.
- Aliniere dinamică a servomotoarelor în funcție de pozițiile detectate.
- Activare secvență de tragere când ținta este aliniată.
- Control complet al modurilor și resetare din tastatură sau buton.

## Funcții cheie implementate

### Pe Arduino:

- `pinModeBare()`, `digitalWriteBare()`, `analogReadBare()` - acces direct la registre.
- `beep(freq, durată)` - feedback sonor fără tone().
- `setRGB()` - stare vizuală în funcție de mod.
- `loop()` - gestionează comenzi seriale + logica modului activ.

### Pe PC (Python):

- `detect\_target()` - caută cercuri și validează poziția.
- `detect\_laser()` - localizează punctul roșu intens.
- `run\_exploration\_mode()` - mișcare servo stânga-dreapta și scanare automată.
- `send(cmd)` - trimite comenzile UART către Arduino cu întârziere minimă controlată.

## Rezultate Obținute



## Performanță:

- Detecție fiabilă a țintei (cerc) și a laserului cu OpenCV.
- Mișcare fluidă a servomotoarelor pentru urmărire pe două axe (pan/tilt).
- Activare precisă a trăgaciului când ținta este centrată.
- Moduri de operare complet funcționale: FREE, SCAN, LOCK, FIRE\_WARN, MANUAL.
- Feedback vizual și sonor prin LED RGB și buzzer.

## Demonstrație video:

► [Vizualizează un Demo](#)

## Concluzii

Proiectul **Sistem Automat de Țintire** a reușit să combine eficient hardware-ul de tip embedded cu procesarea vizuală de pe PC. Am integrat cu succes controlul prin UART, logica de moduri, interfață vizuală (LED RGB), feedback acustic (buzzer), control manual (joystick) și funcționalitate completă de tragere automată.

Pe lângă valoarea educativă și tehnică, proiectul a fost și o provocare practică în a integra mai multe concepte: PWM, UART, ADC, protocoale de comunicație și prelucrare imagine în timp real.

☐ **Notă importantă:** Aproximativ **jumătate din problemele întâlnite** în faza de prototipare s-au rezolvat rapid cu ajutorul **benzii adezive**. Fixările provizorii au devenit permanente, iar improvizația a făcut parte din procesul creativ. ☐

## Download

☐ **Arhivă completă proiect:** [descarcă Auto-Turret-System.zip](#)

☐ **Repository GitHub:** [github.com/AntonescuAlbert/AutoTurret-PM2025](https://github.com/AntonescuAlbert/AutoTurret-PM2025)

## Bibliografie/Resurse

### Resurse Software

- [OpenCV Documentation](#)
- [Arduino Reference](#)
- [PySerial Documentation](#)

## Resurse Hardware

- [Datasheet SG90 Servomotor](#)
- [Datasheet MG996R Servomotor](#)
- [Joystick PS2 Module Pinout](#)
- [Buzzer pasiv - circuit și funcționare](#)

## Alte

- Video tutoriale Arduino + OpenCV de pe YouTube
- Exemple de cod din laboratoarele PM

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/iotelea/albert.antonescu>



Last update: **2025/05/30 12:05**