

Documentatie ESP32 StopWatch

1.Problem Statement

Proiectul propus consta in dezvoltarea unei aplicatii integrate care sa permita utilizatorilor gestionarea unui **cronometru**, a unui **timer**, si al unui **lap counter**, toate accesibile printr-o interfata web. Problemele adresate sunt urmatoarele:

1. Lipsa unui sistem de urmarire a timpului usor accesibil si controlabil de la distanta
2. Nevoia unei interfete prietenoase si responsive, compatibila cu multiple dispozitive (telefon mobil, desktop, tableta)
3. Necesitatea unui design hardware simplu si rentabil pentru implementarea functionalitatilor propuse.

2.The Solution

Proiectul utilizeaza un microcontroller ESP32 configurat ca punct de acces Wi-Fi pentru a oferi o interfata web de control. Solutia implementata include:

- a) Un cronometru capabil sa porneasca, sa se opreasca si sa se reseteze
- b) Un timer configurabil din interfata web cu optiuni predefinite
- c) Un lap counter care numara turele in functie de un interval de tip setat de utilizator

3.Implementation

Flow-ul aplicatiei:

1. Pornirea si configurarea ESP32
 - a. Dupa alimentarea ESP32, acesta initiaza un hotspot Wi-Fi cu un nume si o parola predefinite
 - b. Dispozitivul configureaza un server web local, care gestioneaza cererile utilizatorilor
2. Interactiunea cu interfata web
 - a. Utilizatorii se conecteaza la reseaua creata de ESP32 si acceseaza interfata web printr-un browser
 - b. Interfata e simpla, cu sectiuni dedicate pentru fiecare functionalitate: cronometru, timer si lap counter
 - c. Actualizarile datelor sunt facute automat prin cereri catre server

3. Gestionarea cererilor
 - a. Serverul interpreteaza cererile HTTP trimise de user (ex: "Start", "Stop", "Reset")
 - b. In functie de cererea primita, ESP32 initiaza/opreste cronometru, calculeaza timpul ramas pentru timer sau numara lap-uri bazate pe intervale de timp
4. Sincronizarea timpului
 - a. ESP32 foloseste o functie interna care masoara timpul scurs de la pornire pentru a calcula valori timpul ramas pentru timer
5. Transmiterea datelor in timp real
 - a. Datele actualizate sunt trimise catre interfata in format JSON si afisate utilizatorului

Descrierea functionalitatilor principale

- A. **Cronometru:** pornirea si oprirea cronometrului sunt gestionate intern prin masurarea timpului scurs in milisecunde. La solicitare, timpul este intors spre interfata web in format standard HH:MM:SS
- B. **Timer:** acesta porneste de la o valoare prestabilita de utilizator (in minute) si scade pana la zero. Valoarea ramasa este calculata periodic si trimisa catre interfata
- C. **Lap Counter:** aceasta functionalitate numara intervalele de timp prestabilite. Daca numarul de secunde setat este numarat, variabila ce retine lap counter-ul creste, iar numaratorul este resetat.

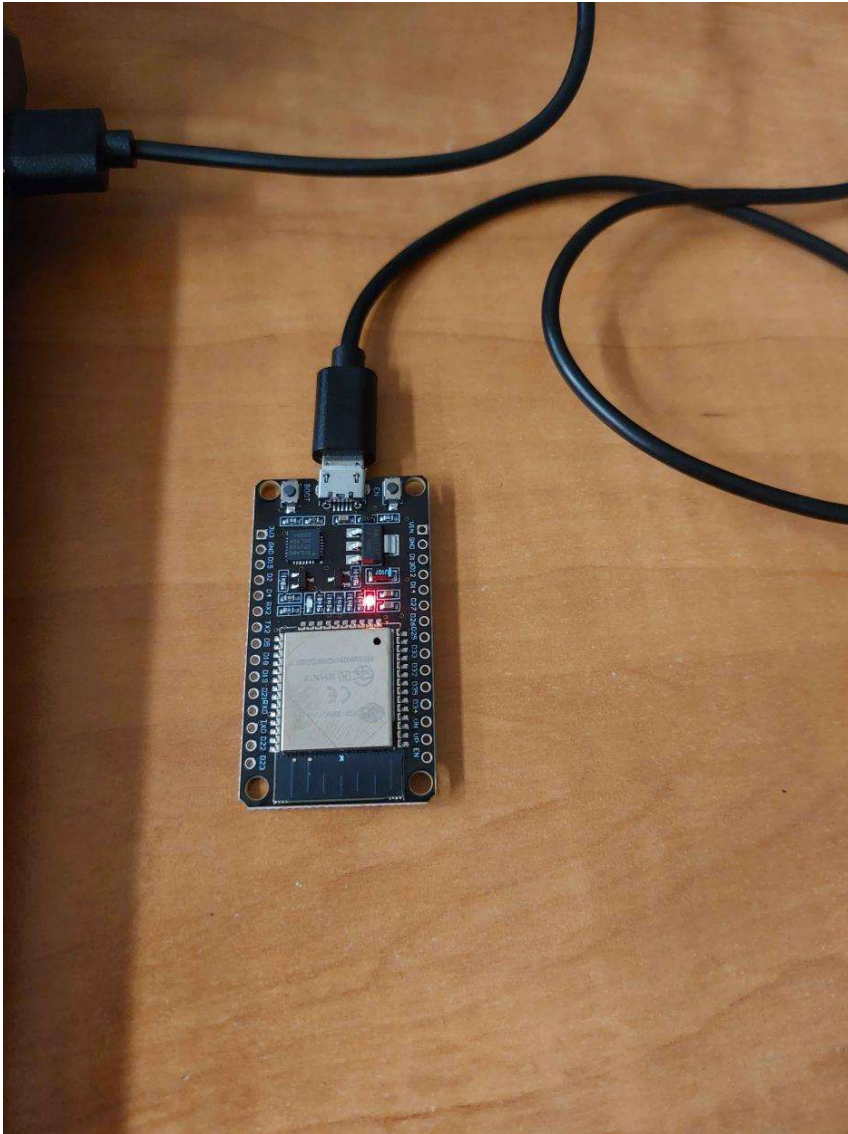
Tehnologii utilizate:

- ✓ **ESP32:** microcontroller cu suport Wi-Fi integrat
- ✓ **Server web integrat:** permite gestionarea cererilor HTTP
- ✓ **HTML, CSS, JavaScript:** creeaza o interfata web simpla, responsive si usor de utilizat pe orice platforma
- ✓ **JSON:** format pentru schimbul de date intre server si client, utilizat pentru transmiterea informatiilor in timp real

4.Circuit Diagram/Schematics

Proiectul utilizeaza doar placa ESP32 Devkit V1, fara componente externe suplimentare. Aceasta este alimentata prin portul USB si functioneaza autonom, datorita functionalitatilor Wi-Fi integrate.

5. Poza cu Proiectul



6. Bibliography

- 1) [DOIT ESP32 DEVKIT V1 Development Board Details, Pinout](#)
- 2) [JavaScript Tutorial](#)
- 3) [HTML Tutorial](#)
- 4) [CSS Tutorial](#)
- 5) [DMP Lab 7 - ESP32 Part 1 - RO - Documente Google](#)
- 6) [DMP Lab 8 - ESP32 Part 2 - RO - Documente Google](#)