

# Lap Timing System

## Introducere

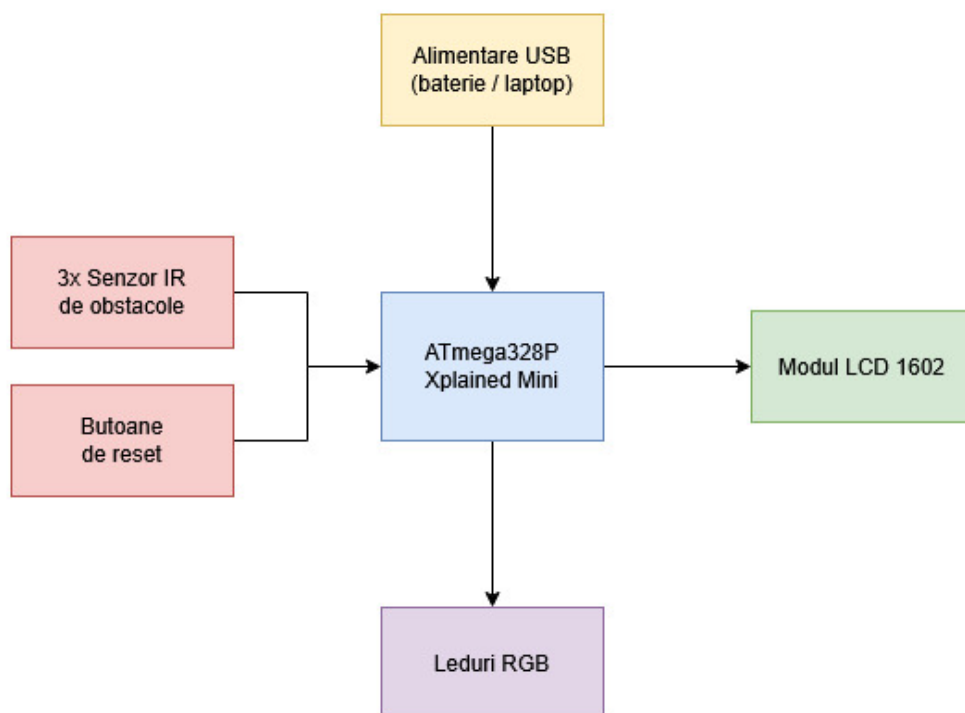
Lap Timing System este un sistem de cronometraj pe sectoare pentru un vehicul care parcurge o pistă închisă, inspirat din modul în care se afișează timpii pe sectoare în calificările de Formula 1. Sistemul măsoară timpul scurs între trecerile vehiculului prin trei puncte de control de pe pistă și oferă atât citiri numerice, cât și un feedback vizual conform convenției folosite de transmisiunile F1.

**Scopul** proiectului este de a transforma rularea pe pistă într-o experiență de tip *time trial*, în care utilizatorul are imediat o evaluare a performanței pe fiecare sector, fără să fie nevoit să compare numere.

### De ce cred că este util:

- E distractiv - adaugă o miză de cronometraj la rularea pe pistă.
- Îți arată unde câștigi și unde pierzi timp pe fiecare sector, nu doar pe turul întreg.
- Poți compara timpii cu prietenii sau să-ți urmărești progresul între sesiuni.

## Descriere generală



Sistemul este construit în jurul plăcii ATmega328P Xplained Mini, alimentată prin USB de la un laptop sau o baterie externă. Microcontrolerul comunică cu patru grupuri de periferice:

**Detectie.** Trei senzori IR de obstacole sunt amplasați la limitele dintre sectoare, lângă pistă. Fiecare senzor declanșează o întrerupere externă pe ATmega328P când vehiculul trece prin dreptul lui.

**Cronometraj.** Timer1 al microcontrolerului (16 biți) servește ca bază de timp principală. La fiecare întrerupere de la senzori, valoarea curentă a timer-ului este capturată în software și folosită pentru a calcula timpul scurs de la trecerea anterioară.

**Feedback vizual.** Trei LED-uri RGB (câte unul per sector) arată rezultatul comparației dintre timpul nou-măsurat, turul anterior și recordul sesiunii:

- **Mov** - record absolut pe sectorul respectiv
- **Verde** - record personal pe sectorul respectiv
- **Galben** - mai lent decât recordul personal

**Afișaj.** Un LCD 16×2 conectat prin I<sup>2</sup>C arată numărul turului curent, timpii pe sectoare în timp real, cel mai bun tur și diferențele față de el.

**Control.** Două butoane permit resetarea sesiunii curente, respectiv ștergerea tuturor valorilor salvate pentru ziua curentă.

Sesiunea funcționează în model *qualifying lap*: cronometrul pornește la prima trecere prin dreptul senzorului de start, iar fiecare trecere ulterioară prin același punct închide turul curent și îl pornește pe următorul.

## Hardware Design

### Listă de componente

Componentă	Rol în proiect
ATmega328P-XMINI	Microcontrolerul care rulează firmware-ul. mEDBG-ul integrat pe placă permite programarea și debugging-ul prin USB.
3× Senzor IR de obstacole (3.3-5 V)	Detectează trecerea vehiculului prin cele trei puncte de control.
3× LED RGB 5 mm cu catod comun	Indicator vizual per sector (mov / verde / galben).
1× LCD 1602 + modul I <sup>2</sup> C	Afișaj numeric pentru tur, sectoare și delta.
2× Buton PCB tactil (12×12×7.3 mm)	Resetare sesiune și resetare valori salvate.

### Justificarea componentelor

**Senzorii IR de obstacole** au fost aleși pentru că oferă un semnal digital simplu (HIGH/LOW), compatibil direct cu pinii cu întrerupere externă ai ATmega328P. Pragul de detecție se reglează din potențiometrul de pe modul, ceea ce permite calibrarea în funcție de pistă și de obiectul detectat.

**LED-urile RGB cu catod comun** au fost preferate în locul a trei LED-uri colorate separate per sector, pentru ca toate cele trei culori să apară în aceeași poziție fizică. Pentru cele trei culori discrete folosite (mov, verde, galben) este suficient controlul digital pe fiecare canal, fără PWM.

**LCD-ul cu adaptor I<sup>2</sup>C** reduce numărul de pini ocupați de la șase (mod paralel) la doi (SDA, SCL), eliberând GPIO pentru senzori, LED-uri și butoane.

## Software Design

**Mediu de dezvoltare:** PlatformIO cu framework-ul MiniCore.

Logica programului se bazează pe măsurarea timpului scurs între întreruperile generate de cei trei senzori IR, folosind Timer1 pe 16 biți ca bază de timp.

### Structura algoritmului:


1. Inițializare: configurarea pinilor I/O, a Timer1, a întreruperilor externe pentru senzori și a interfeței I<sup>2</sup>C pentru LCD.
2. Bucla principală:
  1. Așteptarea unui flag setat de un ISR la trecerea vehiculului prin dreptul unui senzor.
  2. Calculul timpului pe sectorul închis ca diferență între capturile Timer1.
  3. Compararea timpului nou cu recordul personal al sesiunii și cu recordul absolut al zilei.
  4. Actualizarea culorii LED-ului pentru sectorul respectiv (mov / verde / galben).
  5. Actualizarea afișajului LCD cu turul curent, timpii pe sectoare și delta.
  6. Tratarea butoanelor (cu debouncing software) pentru reset sesiune și reset valori salvate.

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## Concluzii

## Download

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

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).  
**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

## Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

## 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/prj2026/bianca.popa1106/toma.ariciu>



Last update: **2026/05/09 21:58**