

Lap Timing System

Introducere

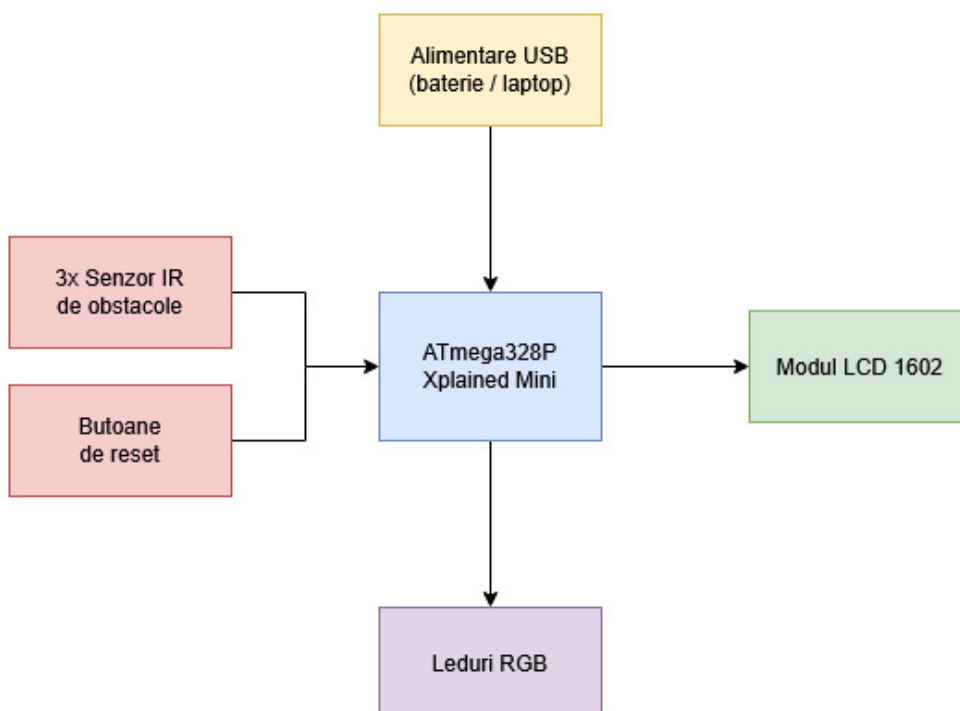
Lap Timing System este un sistem de cronometraj pe sectoare pentru un vehicul care parcurge o pistă închisă, inspirat din modul în care se afișează timpii pe sectoare în calificările de Formula 1. Sistemul măsoară timpul scurs între trecerile vehiculului prin trei puncte de control de pe pistă și oferă 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:

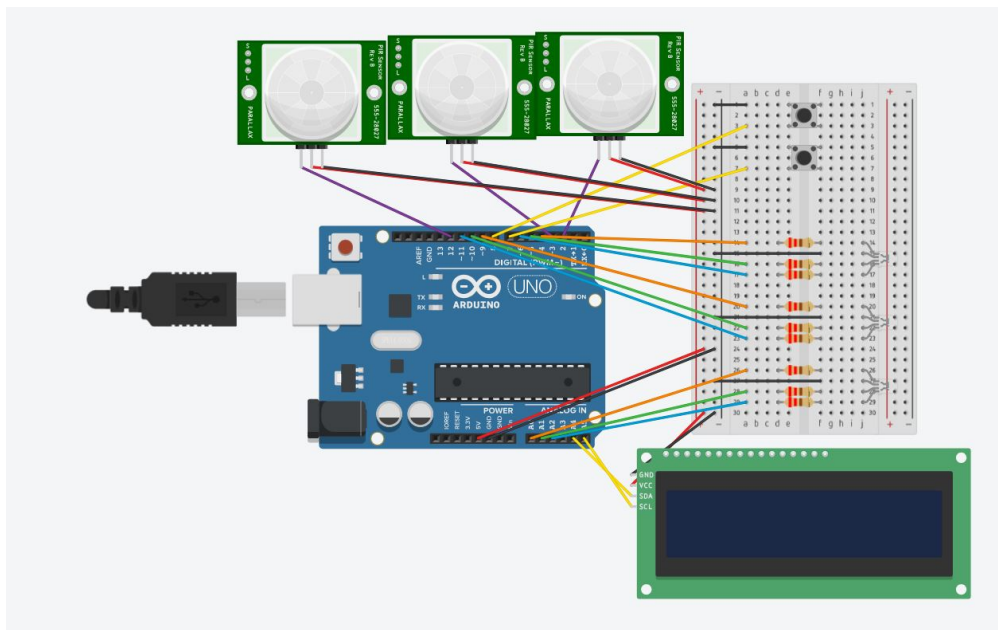
- **Mov** - record absolut pe sectorul respectiv
- **Verde** - record personal pe sectorul respectiv
- **Galben** - mai lent decât recordul personal

Afișaj. Un LCD 16x2 conectat prin I²C 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



BOM:

Componenta	Cantitate
ATmega328P Xplained Mini	1
IR obstacle sensor 3.3-5V	3

LED RGB 5mm 4-pini, catod comun	3
Rezistor 220Ω	9
LCD 1602	1
Modul I ² C LCD interface (PCF8574)	1
Breadboard MB-102 (830 puncte)	1
Buton tactile PCB 12×12×7.3 mm	2

Descrierea componentelor:

- **IR obstacle sensor:** cei 3 senzori detectează trecerea mașinuței prin fiecare sector al pistei. Ieșirea OUT a fiecăruia e conectată la un pin cu capacitate de întrerupere — întreruperi externe (INT) pentru primii doi senzori și pin change interrupt (PCINT) pentru al treilea.
- **LED RGB:** cele 3 LED-uri RGB indică vizual performanța per sector. Combinațiile digitale folosite: mov pentru record absolut, verde pentru record personal, galben pentru tur mai lent.
- **Buton tactil:** butonul 1 resetează sesiunea curentă, butonul 2 resetează valorile salvate. Ambele folosesc pull-up-ul intern al ATmega-ului.
- **LCD 1602 prin I²C:** afișează numărul turului, timpii pe sector, cel mai bun tur și diferențele dintre tururi. Adresa modulului I²C: 0x27.
- **Rezistori 220Ω:** limitează curentul prin fiecare canal al LED-urilor RGB pentru a se încadra în specificațiile ATmega328P (max 40 mA per pin).

Sistemul folosește un ground comun între placă și breadboard, alimentat la 5V prin portul USB al plăcii.

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 I²C 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: **2026/05/11 13:18**