

# SafeDrive: Sistem Auto pentru Detectarea Alcoolului

## Introducere

SafeDrive este un sistem embedded de siguranță conceput pentru a simula funcționalitatea dispozitivelor comerciale de tip Alcolock. Obiectivul său principal este de a evalua concentrația de alcool din aerul expirat de șofer și de a preveni activ pornirea motorului vehiculului dacă limita legală este depășită.

Motivația din spatele acestui proiect este construirea unei aplicații tangibile, menite să salveze vieți, care face legătura între senzorii hardware de bază și sistemele de siguranță activă. Prin renunțarea la mediul standard Arduino, acest proiect servește drept un exercițiu complex de programare AVR. Acesta provoacă dezvoltatorul să orchestreze perfect multiple periferice hardware — cum ar fi ADC pentru procesarea semnalelor analogice, I2C operat manual la nivel de regiștri pentru afișarea datelor, UART Software pentru multimedia și Timerele hardware pentru o arhitectură complet non-blocantă — într-o singură aplicație unitară, bazată pe întreruperi.

## Descriere generală



Întregul flux operațional este orchestrat de microcontrolerul ATmega328P. Arhitectura sistemului este împărțită în patru blocuri logice principale:

- **Pornire bazată pe Întreruperi (Interrupt-Driven):** Interacțiunea începe când utilizatorul apasă butonul de "Contact". În loc să irosească cicluri de procesor verificând constant starea butonului (polling), sistemul se bazează pe o Întrerupere Externă (INT1) pentru a se trezi instantaneu și a declanșa secvența de testare.
- **Procesarea Semnalelor Analogice:** Un senzor de gaz MQ-3 furnizează continuu o tensiune analogică proporțională cu concentrația de alcool către ADC-ul microcontrolerului, valoarea fiind filtrată digital și analizată printr-un algoritm de Peak-Hold timp de 7 secunde.
- **Feedback Vizual Custom în Timp Real:** Stările sistemului, timpul rămas și alcoolemia instantanee sunt trimise către un ecran LCD 20x4. Comunicarea este gestionată eficient prin protocolul I2C folosind un expander PCF8574, condus printr-un driver software scris manual de la zero, fără biblioteci dedicate LCD.
- **Sistem de Alertă Multimedia și Auditiv:** Dacă sistemul detectează un test picat, simulează blocarea motorului (LED Roșu) și declanșează o avertizare. Microcontrolerul comandă via UART (SoftwareSerial) un modul DFPlayer Mini pentru asistență vocală și pulsează asincron un buzzer

activ folosind baza de timp a Timerului 1.

## Hardware Design

Tabelul de mai jos detaliază componentele necesare pentru a construi prototipul SafeDrive:

Nr.	Nume Componentă	Cant.	Rol Tehnic & Detalii
1	ATmega328P Xplained Mini	1	Unitatea centrală de procesare care gestionează toată logica, întreruperile și timpii de execuție.
2	Modul Senzor de Gaz MQ-3	1	Detectează concentrația de alcool; oferă un semnal analogic (0-5V).
3	Display LCD 20x4 + I2C	1	Oferă feedback vizual extins. Modulul PCF8574 atașat reduce cablarea la doar pini SDA și SCL.
4	Modul Audio DFPlayer Mini	1	Acționează ca unitate de decodare și stocare (MicroSD) pentru fișierele audio de ghidare.
5	Modul Amplificator LM386 Mono	1	Prelucrează semnalul de nivel mic (DAC) din playerul audio pentru a ataca eficient difuzorul.
6	Difuzor 8Ω 0.5W	1	Convertește semnalele electrice amplificate în avertizări sonore/vocale.
7	Buton tactil (Push Button)	1	Simulează contactul mașinii, fiind conectat la un pin de întrerupere externă hardware.
8	LED-uri de Status (R, B, V)	3	Indică starea simulată a motorului (Blocat/Avertisment/Deblocat).
9	Buzzer Activ	1	Oferă semnalizare acustică suplimentară (chiuit EKG) la depășirea pragului critic.
10	Breadboard 830 puncte	1	Platforma principală de prototipare pentru rutarea alimentării și a semnalelor.
11	Fire conexiune DuPont	1 set	Folosite pentru interconectarea modulelor cu placa Xplained Mini.

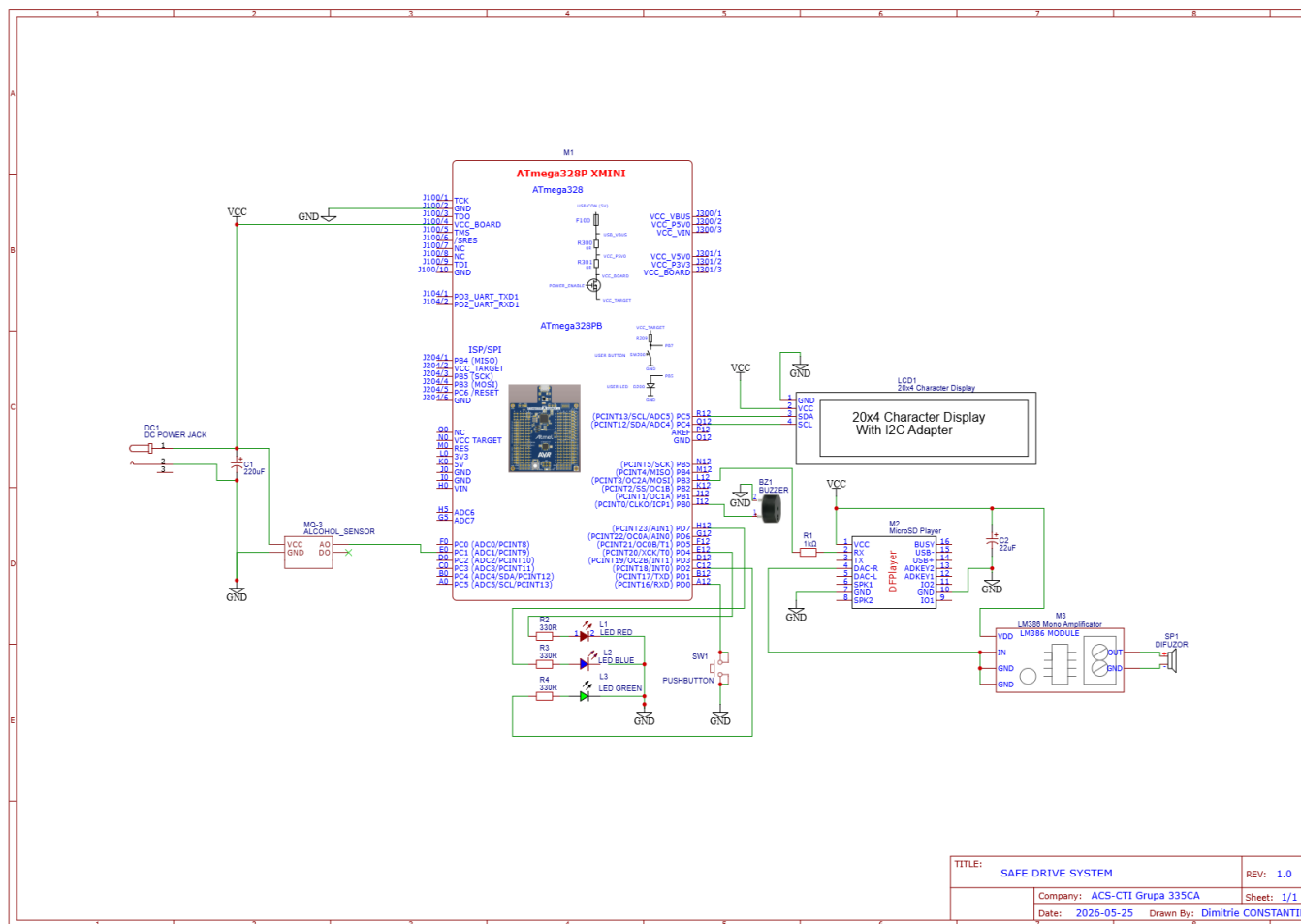
## Alocarea Pinilor (Pinout) și Justificare

Alocarea pinilor nu a fost aleatorie, ci a respectat strict arhitectura internă a microcontrolerului ATmega328P:

- **Ecran LCD (I2C):** Sunt folosiți pini **PC4 (SDA)** și **PC5 (SCL)**. Aceștia sunt pini hardware dedicați magistralei TWI (Two-Wire Interface).
- **Modul Audio DFPlayer (UART):** Sunt utilizați pini **PB2 (D10 - RX)** și **PB3 (D11 - TX)** emulați prin SoftwareSerial, lăsând portul hardware UART nativ liber pentru debugging pe PC.
- **Senzor MQ-3 (Analog Out):** Este conectat la **PC1 (ADC1)**, un canal direct al convertorului Analog-Digital intern.
- **LED-uri RGB și Buzzer:** Sunt conectate la pini GPIO standard (**PD2, PD4, PD7** pentru LED-uri; **PB0** pentru buzzer), manipulate direct prin bitmasking pe regiștri.

- **Buton (Contact):** Este conectat la **PD3 (INT1)**, permițând trezirea și comutarea de stare instantanee exclusiv prin vectorul de întrerupere hardware pe front căzător.

## Explicația Schemei Electrice



- **Magistralele de alimentare:** Toate modulele împart o alimentare comună. Magistrala de +5V și cea de GND sunt distribuite printr-un breadboard de la pinii de putere ai plăcii Xplained Mini, având un condensator electrolitic principal de filtrare pentru absorbția vârfurilor de curent generate de amplificatorul audio.
- **Protecția logică:** Butonul de contact utilizează rezistența de pull-up internă activată din cod pentru a asigura un nivel logic HIGH stabil atunci când nu este apăsat. LED-urile sunt protejate de rezistențe limitatoare de curent de 330Ω pe fiecare canal.
- **Filtrarea Audio:** Semnalul audio curat este extras din pinul DAC (neamplificat) al modulului DFPlayer și rutat direct către intrarea modulului LM386, asigurând un sunet puternic și clar pe difuzorul de 8Ω.

## Software Design

Firmware-ul pentru SafeDrive este scris integral în C, manipulând direct regiștrii AVR pentru a asigura o performanță maximă și un consum minim de memorie (SRAM).

- **IDE & Toolchain:** Visual Studio, PlatformIO
- **Logica Principală de Implementare:**
- **Baza de Timp Asincronă:** S-a utilizat Timer-ul 1 în mod CTC pentru a genera un "tick" exact la fiecare 500ms. Acesta guvernează decrementarea ferestrei de test (7 secunde), eșantionarea pe senzor și pulsarea buzzer-ului, lăsând funcția `loop()` complet non-blocantă.
- **Mașină de Stări (FSM):** Sistemul funcționează pe baza unei arhitecturi finite-state-machine, fiecare stare controlând simultan LCD-ul, LED-urile, buzzer-ul și feedback-ul audio.
- **Comunicare I2C Low-Level:** Driverul pentru LCD a fost implementat manual prin manipularea directă a registrelor TWI, fără utilizarea bibliotecilor standard Arduino.

## Explicarea calibrării elementelor de senzorică

Calibrarea senzorului analogic de gaz MQ-3 (bazat pe dioxid de staniu - SnO<sub>2</sub>) a necesitat o abordare empirică, divizată în două etape esențiale datorită naturii termochimice a componentei:

- **Pre-încălzirea (Burn-in hardware):** Deoarece senzorul funcționează pe baza unui filament de încălzire intern, acesta necesită o perioadă de stabilizare. La prima utilizare, componenta a fost menținută alimentată timp de 24 de ore în aer curat pentru a se "arde" pelicula internă. Suplimentar, la fiecare testare curentă este prevăzut un timp de caldaj (warm-up) de 5-10 minute.
- **Determinarea pragului de repaus (Baseline) și conversia:** Prin monitorizarea valorilor brute via portul Serial într-un mediu cu aer curat, s-a determinat o constantă de bază a mediului de aprox. 120 (stocată în variabila `prag_aer_curat`). Pentru a obține o estimare a concentrației de alcool, sistemul calculează diferența dintre valoarea curentă citită și acest prag de repaus, aplicând ulterior un factor de atenuare și o conversie liniară empirică:

$$\text{BAC}(\%) = (\text{Valoare\_ADC} - \text{Prag\_AerCurat}) * 0.00025$$

Orice valoare a senzorului care fluctuează sub pragul de bază este ignorată, fiind forțată la 0.00% pentru a preveni declanșările accidentale din cauza zgomotului termic sau a fluctuațiilor de tensiune.

## Optimizări software (Cum, De ce, Unde)

Pentru a asigura rularea stabilă a proiectului pe un microcontroler cu resurse limitate (ATmega328P dispune de doar 2KB SRAM), au fost implementate două optimizări majore:

### 1. Optimizarea memoriei dinamice (Prevenirea Memory Leaks / Heap Fragmentation)

- **Cum:** Au fost eliminate complet instanțierile de obiecte de tip `String()` din rutinele de logging serial. Concatenările costisitoare au fost înlocuite cu apeluri succesive, bazate pe text constant,

folosind `Serial.print()`. În plus, printarea datelor de diagnoză a fost mutată strict în interiorul blocului condiționat de flag-ul Timerului 1.

- **De ce:** Alocarea și dealocarea dinamică repetată la viteze mari ducea la fragmentarea rapidă a memoriei SRAM. Această epuizare a resurselor bloca procesorul într-o buclă de așteptare, provocând "înghețarea" sistemului.
- **Unde:** În funcțiile `loop()` și `procesare_date_senzor()`.

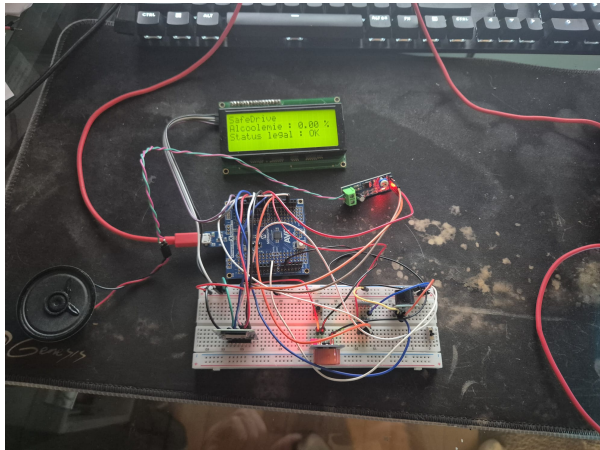
## 2. Filtru digital de netezire (Low-Pass Filter / Exponential Moving Average)

- **Cum:** A fost implementat un filtru matematic software pe eșantioanele ADC citite de senzor, folosind următoarea ecuație:

$$V_{\text{filtrat}} = (V_{\text{vechi}} * 0.8) + (V_{\text{nou}} * 0.2)$$

- **De ce:** Senzorul MQ-3 are o reacție termochimică violentă la contactul cu surse concentrate de vapori. Filtrul implementat absoarbe doar 20% din valoarea nouă la fiecare iterație (500ms), netezind curba de răspuns și permițând trecerea fluidă prin stările de alarmă (Atenție → Interzis).
- **Unde:** La fiecare bătaie a Timerului 1 în `loop()`.

## Stadiul actual al implementării și Testare Hardware



```
arduino$ analogread A1
A1 = 256
arduino$ analogread A1
A1 = 339
arduino$ analogread A1
A1 = 325
arduino$ analogread A1
A1 = 593
arduino$ analogread A1
A1 = 584
```

Sistemul SafeDrive este complet funcțional. Montajul final integrează stabil modulele pe breadboard, iar schema electrică este proiectată și validată. Testele efectuate demonstrează capacitatea sistemului de a prelua date analogice corecte de la senzor și de a reacționa în timp real la pragurile de alcool setate.

Prin intermediul unei interfețe shell de test, comanda `analogread A1` validează atât alimentarea senzorului MQ-3, cât și integritatea magistralei ADC interne a microcontrolerului.

## Motivarea alegerii bibliotecilor

Pentru a menține codul modular și eficient, au fost utilizate atât biblioteci native AVR, cât și implementări software custom:

- `<avr/io.h>` și `<avr/interrupt.h>`: Biblioteci native AVR utilizate pentru manipularea directă a registrelor și gestionarea întreruperilor hardware.
- `<util/delay.h>`: Utilizată exclusiv în secvențele critice de inițializare hardware unde sunt necesare întârzieri precise.
- **Driver I2C Custom**: Comunicarea cu LCD-ul prin PCF8574 este realizată fără biblioteca `Wire.h`, folosind control direct asupra perifericului TWI pentru performanță și înțelegere low-level completă.
- `<SoftwareSerial.h>`: Folosită pentru emularea unei interfețe UART dedicate DFPlayer Mini, păstrând UART-ul hardware liber pentru debugging.
- `<DFRobotDFPlayerMini.h>`: Simplifică trimiterea comenzilor seriale către DFPlayer pentru controlul redării audio și al volumului.

## Elementul de noutate al proiectului

Elementul de noutate constă în integrarea unui sistem de alertă multi-senzorial, adaptiv și complet asincron, capabil să combine feedback-ul vizual, auditiv și vocal într-o singură arhitectură embedded de dimensiuni reduse.

Spre deosebire de etilotestele comerciale clasice bazate exclusiv pe praguri fixe, SafeDrive implementează:

- filtrare digitală software în timp real;
- analiză Peak-Hold pe fereastră temporală;
- feedback vocal prin DFPlayer;
- semnalizare EKG prin buzzer asincron;
- arhitectură complet event-driven bazată pe întreruperi și FSM.

Aceste funcționalități permit o experiență mult mai realistă și robustă, apropiată de sistemele embedded automotive moderne.

## Justificarea utilizării funcționalităților din laborator

Proiectul integrează masiv conceptele low-level studiate la laborator:

- **Laboratorul I/O (Operații pe biți):** LED-urile și buzzer-ul sunt controlate prin manipularea directă a registrelor DDRx și PORTx folosind bitmasking.
- **Laboratorul ADC:** Conversia analog-digitală pentru MQ-3 este configurată manual prin registrele ADMUX și ADCSRA, eliminând funcția `analogRead()`.
- **Laboratorul Timere și Întreruperi:** Timer-ul 1 este configurat în modul CTC pentru generarea unei baze de timp precise și complet non-blocante.
- **Laboratorul UART:** Comunicarea serială cu DFPlayer Mini utilizează UART software pentru transmisia comenzilor audio.
- **Laboratorul I2C/TWI:** Comunicarea cu LCD-ul prin expandorul PCF8574 este realizată prin control direct al magistralei TWI hardware.

## Explicarea scheletului proiectului și Validarea

### Arhitectura interacțiunii:

Scheletul aplicației este complet event-driven. În funcția `setup()` sunt inițializate perifericele hardware, magistralele de comunicație și Timer-ul 1.

Funcția principală `loop()` rămâne permanent non-blocantă și execută doar sarcini dependente de flag-uri software ridicate de ISR-uri.

Fluxul de execuție:

1. Utilizatorul apasă butonul de contact.
2. ISR-ul asociat INT1 declanșează secvența de testare.
3. Timer-ul 1 generează tick-uri periodice la 500ms.
4. ADC-ul citește valoarea senzorului MQ-3.
5. Filtrul EMA netezește semnalul analogic.
6. Algoritmul Peak-Hold reține valoarea maximă pe durata ferestrei de 7 secunde.
7. FSM-ul decide starea sistemului.
8. LCD-ul, LED-urile, DFPlayer-ul și buzzer-ul sunt actualizate asincron.

Această arhitectură permite funcționarea simultană a tuturor modulelor fără utilizarea funcțiilor blocante de tip `delay()`.

## Concluzii

Dezvoltarea proiectului SafeDrive a reprezentat o provocare inginerească complexă, fiind o tranziție reușită de la utilizarea librărilor generice la controlul hardware de nivel scăzut.

Principala dificultate a fost gestionarea resurselor limitate de memorie și atenuarea zgomotului indus de comutația perifericelor în semnalul audio, problemă rezolvată prin optimizarea rutinei de redare și prin separarea logică a task-urilor asincrone.

În final, proiectul a consolidat competențele de:

- programare AVR low-level;
- utilizare registre hardware;
- configurare Timere în mod CTC;
- dezvoltare FSM;
- programare interrupt-driven;
- filtrare digitală software;
- integrare multi-periferic într-un sistem embedded real-time.

SafeDrive demonstrează succesul integrării unui sistem multi-senzorial robust, capabil să ia decizii critice în timp real fără a necesita un sistem de operare dedicat.

## Download

**Cod Sursă (GitHub):** <https://github.com/DimiConstantin/SafeDrive-System.git>

## Videoclip Demo

[https://drive.google.com/file/d/1NgjwWxK3g2bzu4fEecLFvrpopXJ86x9d/view?usp=drive\\_link](https://drive.google.com/file/d/1NgjwWxK3g2bzu4fEecLFvrpopXJ86x9d/view?usp=drive_link)

## Jurnal

- **Săptămâna 1-2:** Documentare, configurare toolchain PlatformIO și testare hardware minimă (LED + Buton).
- **Săptămâna 3-4:** Implementare driver I2C custom pentru LCD și integrare senzor MQ-3 (calibrare baseline).
- **Săptămâna 5-6:** Dezvoltare mașină de stări (FSM) și implementare logică bazată pe Timer 1 (asincronism).
- **Săptămâna 7-8:** Integrare module audio (DFPlayer + Amplificator) și debug pe partea de zgomot digital / ground loops.
- **Săptămâna 9-10:** Finalizare schemă electrică EasyEDA, optimizare SRAM și redactare

documentație finală.

## Bibliografie/Resurse

- Microchip ATmega328P - Official 8-bit AVR Microcontroller Datasheet.
- DFRobot DFPlayer Mini MP3 Module Application Manual & Specification.
- WaveShare PCF8574 I2C Expander Core Documentation.
- Datasheet LM386 Low Voltage Audio Power Amplifier.
- Note de curs și laborator - Proiectarea cu Microcontrolere (PM), Facultatea de Automatică și Calculatoare, UPB.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/vlad.radulescu2901/dimitrie.constantin>



Last update: **2026/05/25 14:04**