

# Alcohol-Lock

## Introducere

Proiectul **Alcohol-Lock** este un sistem de siguranță activă destinat vehiculelor, conceput pentru a condiționa pornirea motorului de starea de sobrietate a șoferului. Dispozitivul măsoară concentrația de alcool din aerul expirat, procesează datele în timp real și acționează un mecanism de blocare simulat, asigurând în același timp stocarea istoricului de testări pe un suport extern.

Sistemul vizează eliminarea erorii umane în autoevaluarea capacității de condus, oferind o barieră tehnică obiectivă împotriva condusului sub influența alcoolului.

## Descriere generală

Dispozitivul integrează module de achiziție date, control și stocare pentru a îndeplini următoarele funcții:

- **Monitorizare:** Detectarea precisă a vaporilor de alcool folosind un senzor electrochimic.
- **Control Interlock:** Blocarea sau deblocarea contactului auto (simulată prin servomotor) în funcție de pragul legal setat.
- **Avertizare:** Semnalizarea vizuală și acustică a stării sistemului și a tentativelor de pornire neautorizate.
- **Data Logging:** Înregistrarea fiecărei măsurători pe un card microSD pentru trasabilitatea utilizării vehiculului.

## Schemă Bloc



## Hardware Design

### Componente hardware

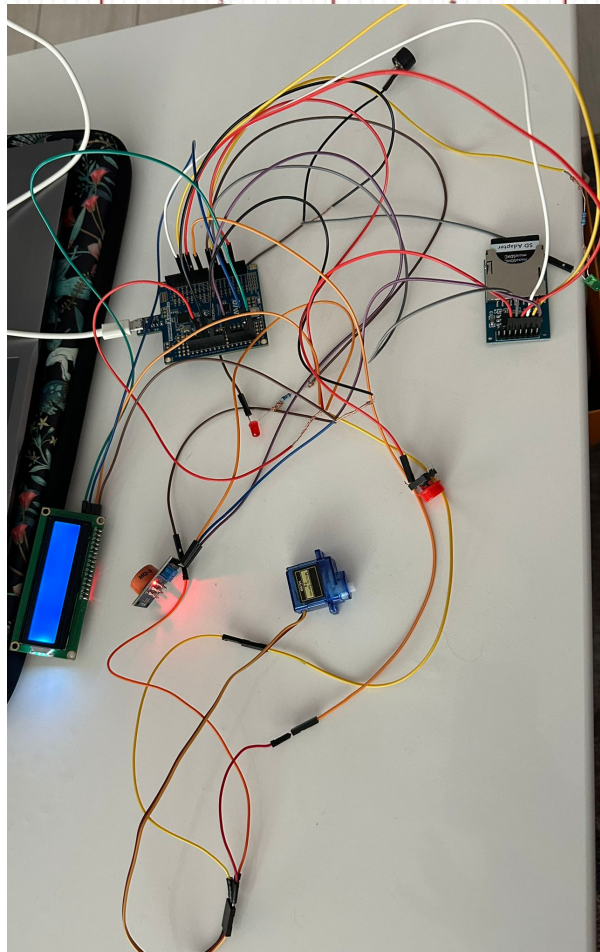
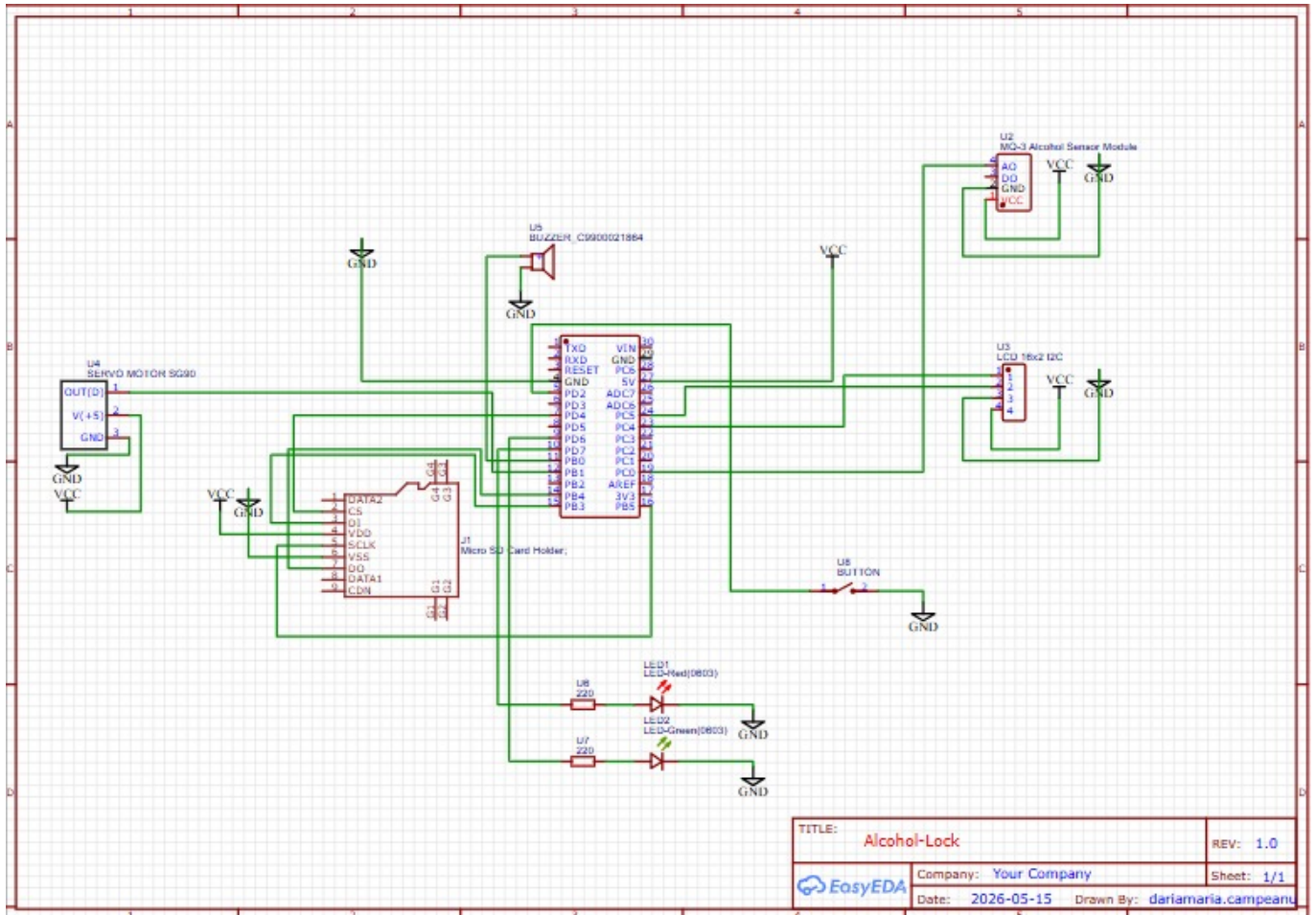
Componentă	Cantitate	Scop
Microcontroler <b>ATmega328P</b>	1	Unitatea centrală de procesare și control
Senzor Alcool <b>MQ-3</b>	1	Achiziția semnalului analogic pentru concentrația de alcool

<b>Modul cititor Card SD</b>	1	Interfață pentru salvarea datelor pe suport extern
<b>Servomotor SG90</b>	1	Simulare mecanică pentru blocarea contactului (PWM)
<b>LCD 16x2 cu adaptor I2C</b>	1	Afișarea concentrației și a mesajelor de stare
<b>Buzzer Pasiv</b>	1	Semnalizare acustică a pragurilor critice
<b>LED-uri (Roșu/Verde)</b>	2	Indicarea vizuală a permisiunii de pornire
<b>Buton Push</b>	1	Declanșarea manuală a procesului de testare
<b>Breadboard Full-size</b>	1	Suport pentru realizarea montajului experimental

## Pinii folosiți

Componentă	Pin ATmega328P	Tip semnal	Justificare
Senzor MQ-3	PC0 (A0)	Intrare analogică	PC0 este conectat la ADC0; semnalul MQ-3 este o tensiune continuă 0-5V proporțională cu concentrația de alcool
Buton START	PD2	Intrare digitală (INT0)	PD2 este legat la întreruperea externă INT0, permițând detecția apăsării fără polling continuu
LED Verde	PD6	Ieșire digitală	Pin digital setat ca ieșire; controlează starea LED-ului verde
LED Roșu	PD7	Ieșire digitală	Pin digital setat ca ieșire; controlează starea LED-ului roșu
Buzzer Pasiv	PB0	Ieșire digitală	Pin digital configurat ca ieșire pentru activarea/dezactivarea directă a buzzerului
Servomotor SG90	PB1 (OC1A)	PWM (Timer1, 16 biți)	Servomotorul necesită PWM la 50 Hz; PB1 este conectat la canalul OC1A al Timerului 1 pe 16 biți
LCD SDA	PC4 (SDA)	I2C date	Pin hardware pentru linia de date SDA a interfeței TWI interne
LCD SCL	PC5 (SCL)	I2C ceas	Pin hardware pentru linia de ceas SCL a interfeței TWI interne
SD Card CS	PB2	Ieșire digitală (SPI CS)	Chip Select pentru perifericul SPI, controlat software
SD Card MOSI	PB3 (MOSI)	SPI date ieșire	Master Out Slave In - Linia hardware de transmisie SPI
SD Card MISO	PB4 (MISO)	SPI date intrare	Master In Slave Out - Linia hardware de recepție SPI
SD Card SCK	PB5 (SCK)	SPI ceas	Serial Clock - Ceasul hardware generat de master pentru magistrala SPI

## Schema electrică



Export to PDF

# Software Design

## Stadiul actual al implementării software

Proiectul software este complet funcțional și structurat modular pe straturi de abstractizare hardware:

- **adc.c / adc.h**: Inițializare și citire ADC pentru senzorul MQ-3.
- **button.c / button.h**: Gestionarea butonului pe pinul PD2 folosind întreruperea externă INT0.
- **lcd\_i2c.c / lcd\_i2c.h**: Driver I2C scris direct pe registre pentru ecranul LCD.
- **servo.c / servo.h**: Controlul servomotorului prin generare de semnal PWM la 50Hz folosind Timer1.
- **sd\_logger.c**: Interfațarea cu sistemul de fișiere Petit FatFs pentru înregistrarea istoricului de măsurători.
- **spi.c / spi.h**: Driverul pentru comunicația SPI hardware utilizat de cardul SD.
- **uart.c / uart.h**: Configurarea transmisiei seriale și redirectarea fluxului standard stdout.

## Motivarea alegerii bibliotecilor folosite

Pentru a eficientiza spațiul redus de memorie al microcontrolerului ATmega328P, s-au scris drivere directe pe registre și s-au adăugat exclusiv componentele software strict necesare:

- **Petit FatFs (R0.02)**: S-a ales varianta extrem de simplificată dezvoltată de ChaN pentru sistemul de fișiere FAT. Aceasta nu utilizează alocare dinamică în memoria RAM și scrie direct în blocul de sectoare deja existente ale fișierului LOG.TXT, reducând masiv amprenta de memorie.
- **avr/io.h & avr/interrupt.h**: Bibliotecile standard din avr-libc folosite pentru maparea regiștrilor specifici microcontrolerului și manipularea vectorilor de întreruperi.
- **util/delay.h**: Utilizată pentru temporizările precise necesare la inițializarea ecranului LCD și a pașilor din mașina de stări.

## Elementul de noutate al proiectului

Proiectul realizează un dispozitiv de siguranță autonom bazat pe Edge Processing. Calculele matematice pentru determinarea concentrației clinice (mg/L) din rezistența brută a senzorului sunt executate local, în timp real, direct pe cip. Totodată, componenta de Data Logger securizează utilizarea vehiculului printr-o jurnalizare continuă pe card, formând o barieră tehnică ce previne pornirea neautorizată a motorului.

## Justificarea utilizării funcționalităților din laborator

Aplicația integrează organic cunoștințele acumulate de-a lungul celor 7 laboratoare ale disciplinei:

**• Laboratorul GPIO**

- Configuarea direcției pinilor prin registrele DDR și controlul stărilor logice prin registrele PORT. A fost esențial pentru gestionarea LED-urilor de stare (Roșu/Verde) și activarea/dezactivarea buzzerului.

**• Laboratorul UART**

- Configurarea baud-rate-ului, activarea transmisiei și maparea fluxului `stdout` către registrul `UDR0`. Acest lucru a permis utilizarea funcției standard `sprintf` pentru formatarea mesajelor, fiind esențial în faza de depanare a algoritmului de calcul.

**• Laboratorul Întreruperi**

- Configurarea registrului `EICRA` pentru detecția pe front căzător (ISC01) și activarea măștii `INT0` din `EIMSK`. Acest laborator este critic deoarece permite preluarea promptă a acțiunii utilizatorului prin ISR(INT0\_vect) fără a încălca procesorul cu bucle de polling.

**• Laboratorul Timere. PWM**

- Configurarea Timerului 1 hardware în mod Fast PWM sau Phase Correct pe 16 biți, definind perioada totală prin registrul `ICR1 = 40000` și factorul de umplere prin `OCR1A`. Este vital pentru generarea semnalului de comandă la frecvența fixă de 50Hz cerută de servomotorul SG90.

**• Laboratorul ADC**

- Activarea convertorului prin bitul `ADEN` și setarea unui prescaler la 128 pentru o conversie stabilă la frecvența de 16MHz a ceasului. Permite citirea semnalului analogic brut furnizat de senzorul MQ-3 pe canalul ADC0.

**• Laboratorul SPI**

- Configurarea pinilor MOSI, SCK și SS ca ieșiri, trecerea modulului SPI în mod Master și setarea divizorului de viteză din registrul `SPCR`. Această funcționalitate este fundamentul pe care rulează driverul de card SD (`sd.c`), permițând transferul rapid de blocuri de date.

**• Laboratorul I2C**

- Configurarea reguștrilor `TWSR` și `TWBR` pentru stabilirea frecvenței de ceas la 100kHz și gestionarea biților de START/STOP/WRITE prin bitul `TWINT`. Permite scrierea datelor către expandorul PCF8574 atașat ecranului LCD, reducând numărul de pini folosiți.

## Scheletul proiectului și validarea

Aplicația funcționează sub forma unei mașini de stări rulate într-o buclă infinită:

- **1. Așteptare:** Sistemul stă în repaus și afișează instrucțiunile inițiale pe LCD până când rutina de întrerupere de la buton modifică flag-ul intern.
- **2. Citire și Procesare:** Se prelevează valoarea analogică ADC și se transformă prin calcul în alcoolemie exprimată în mg/L.
- **3. Salvare:** Datele finale (valoarea ADC brută și cea calculată) sunt înregistrate imediat pe cardul SD în fișierul LOG.TXT.
- **4. Decizie Interlock:** Dacă alcoolemia depășește pragul de 0.15 mg/L, servomotorul rămâne blocat, se aprinde LED-ul Roșu și buzzerul sună alert. Dacă șoferul este sub prag, se aprinde LED-ul Verde, iar servomotorul deblochează contactul.

Validarea s-a făcut prin monitorizarea mesajelor de debug pe interfața UART și prin citirea ulterioară a cardului SD pe un PC pentru a asigura scrierea corectă a datelor text.

## Demo Video

Demonstrația practică a funcționării sistemului în ambele cazuri de testare:

### [Demo Video Alcohol-Lock - YouTube](#)

- **Cazul Sobru:** Se detectează o concentrație sigură (0.00 mg/L), se aprinde LED-ul Verde și servomotorul deblochează contactul.
- **Cazul Alcool detectat:** La apropierea vaporilor de alcool, ecranul afișează valoarea depășită și alerta "Nu poti conduce!", fiind acompaniată de aprinderea LED-ului Roșu și sunetul de avertizare al buzzerului.

## Calibrarea senzorului

Senzorul MQ-3 își modifică rezistența internă la contactul cu alcoolul. Calibrarea s-a făcut folosind valoarea determinată experimental în aer curat  $R_0 = 60.0 \text{ k}\Omega$  și o rezistență de sarcină  $R_L = 10.0 \text{ k}\Omega$ :

- Tensiunea de ieșire se obține din raportul valorii brute ADC raportat la 1024 pași.
- Se determină rezistența curentă a senzorului  $R_s$  pe baza divizorului de tensiune.
- Concentrația finală în mg/L se calculează pe baza raportului dintre  $R_s$  și  $R_0$  folosind coeficienții curbei logaritmice din fișa tehnică.
- Valorile reziduale mici, sub 1.00 mg/L, sunt interpretate ca fiind 0.0 mg/L pentru a elimina zgomotul de fond.

## Optimizări software realizate

- **Oprirea PWM-ului la repaus:** În `servo.c`, funcția `servo\_lock` decuplează semnalul PWM din registre și pune pinul PB1 direct pe starea LOW. Acest lucru elimină vibrațiile parazite ale servomotorului și scade consumul de curent când nu se face mișcare.
- **Eliminarea operanzilor float din sprintf:** Pentru a salva spațiu prețios în memoria Flash, valoarea float a fost extrasă prin conversie directă în două variabile întregi separate (partea întreagă și partea zecimală), salvând peste 15% din spațiul de stocare al programului.
- **Fortărea salvării pe card (Flush):** Imediat după scrierea textului, se apelează funcția de încheiere cu parametri nuli din Petit FatFs. Acest lucru golește bufferul RAM direct în memoria nevolatilă a cardului, securizând logurile în cazul unei deconectări bruște a bateriei vehiculului.

## Cod Relevant

Mai jos este prezentată funcția principală care orchestrează algoritmul de prelevare, calculul alcoolemiei, scrierea pe cardul SD și decizia de blocare a contactului:

```
float calculeaza_alcoolemie(uint16_t adc_raw) {  
    if (adc_raw == 0) adc_raw = 1;
```

```
float v_out = ((float)adc_raw / 1024.0) * 5.0;
float rs = RL_VALUE * (5.0 - v_out) / v_out;
float ratio = rs / R0_CLEAN_AIR;
float mgL = 0.4 * pow(ratio, -1.431);
if (mgL < 1.00) { mgL = 0.0; }
return mgL;
}
```

Rutina de tratare a întreruperii externe asociată butonului START:

```
ISR(INT0_vect) {
    button_flag = 1;
}
```

## Repozitor GitHub

Întregul cod sursă al proiectului, driverele dezvoltate și fișierele de configurare PlatformIO pot fi consultate la următorul link:

- <https://github.com/dariacampeanu/Alcohol-Lock>

## Bibliografie

- **Chan**, \*Petit FatFs Module User Application Interface\*, disponibil online la: <http://elm-chan.org/fsw/ff/00pff.html>.
- **Microchip Technology**, \*ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash - Datasheet\*, <https://www.microchip.com>.
- **Hanwei Electronics Co., Ltd**, \*MQ-3 Gas Sensor for Alcohol Detection - Technical Data\*, Product Datasheet.
- **Librăria avr-libc**, \*AVR Libc Reference Manual\*, disponibil online la: <https://www.nongnu.org/avr-libc/>.

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/vlad.radulescu2901/daria.campeanu>



Last update: **2026/05/25 08:50**