

# Breathalyzer (Etilotest)

## Introducere

**What it does:** The device measures the alcohol concentration in a person's exhaled breath using an MQ-3 gas sensor and displays the result on an LCD/OLED screen, signaling the status via LEDs and an optional buzzer.

**Purpose:** To provide an accessible, low-cost sobriety test tool built entirely from scratch using ATmega328P peripherals.

**Starting idea:** The project originated from the practical need for a simple, register-level embedded systems project that integrates multiple lab concepts into a real-world product.

**Why it is useful:** It demonstrates a complete embedded pipeline — analog sensing, interrupt-driven input, I<sup>2</sup>C communication, and GPIO output — while producing a functional, demonstrable device.

## Descriere generală

The diagram below shows all hardware and software modules and how they interact.



### Module descriptions:

- **MQ-3 sensor** — outputs an analog voltage proportional to alcohol concentration → read via ADC (PA0)
- **Temperature sensor** (optional) — provides ambient temperature for MQ-3 calibration → read via ADC (PA1)
- **Buttons x2** — trigger test start and calibration → connected to INT1 (PD3) and INT2 (PB2) with software debouncing via systicks
- **LEDs (green / yellow / red)** — indicate safe / warning / danger status → controlled via GPIO (PC0–PC2)
- **LCD/OLED display** — shows measured BAC value and status text → driven via I<sup>2</sup>C (TWI)
- **Buzzer** (optional) — auditory alert on high readings → controlled via PWM or GPIO
- **ATmega324P** — central MCU coordinating all peripherals

# Hardware Design

## Bill of materials:

Component	Qty	Notes
—	—	—
ATmega324P	1	Main MCU
MQ-3 alcohol sensor	1	Analog output
Temperature sensor (LM35 or NTC)	1	Optional, for calibration
Push buttons	2	Test start + calibration
LEDs (green, yellow, red)	3	Status indicators
220Ω resistors	3	LED current limiting
LCD 16×2 or OLED 128×64 (I <sup>2</sup> C)	1	Display module
Buzzer	1	Optional audio alert
Breadboard + jumper wires	—	Prototyping
5V USB power supply or battery	1	Power
16 MHz crystal + 2×22pF caps	1	MCU clock
10kΩ pull-up resistors	2	I <sup>2</sup> C lines SDA/SCL

## Pin mapping:

- MQ-3 analog output → PA0 (ADC channel 0)
- Temperature sensor → PA1 (ADC channel 1)
- Button 1 → PD3 (INT1)
- Button 2 → PB2 (INT2)
- LED green → PC0, LED yellow → PC1, LED red → PC2
- SDA → PC1 (TWI), SCL → PC0 (TWI)
- Buzzer → OC0B (PD5) or PD7

## Diagram

# Software Design

**Development environment:** AVR-GCC + avrdude / Microchip Studio

## Third-party libraries and sources:

- No external libraries — all peripheral drivers written from scratch using registers
- Optional: lightweight I<sup>2</sup>C LCD driver adapted from course lab materials

## Planned algorithms and data structures:

- **ADC sampling** — single-shot mode on PA0/PA1; average of 16 samples per reading for noise reduction
- **MQ-3 calibration** — baseline resistance R0 measured in clean air on startup; BAC estimation uses the Rs/R0 ratio from the MQ-3 datasheet characteristic curve
- **Temperature compensation** — Rs/R0 ratio adjusted using ambient temperature reading from PA1
- **Interrupt handling (INT1/INT2)** — button press sets a volatile flag inside the ISR; debounce enforced via systicks counter (Timer1 CTC, 1 ms tick); flag processed in main loop
- **Display logic** — BAC classified into ranges (< 0.2‰ → green, 0.2–0.5‰ → yellow, > 0.5‰ → red); value and status string sent over I<sup>2</sup>C to display
- **LED/Buzzer output** — GPIO write for LEDs; optional PWM tone on OC0B for buzzer alert

## Rezultate Obținute

- Clean air → 0.000 mg/L
- Inside an alcohol recipient → >0.2mg/L

## Concluzii

To be completed ===== Download ===== <note warning> \* Github:

[https://github.com/Mihai4GH/PM\\_Proiect](https://github.com/Mihai4GH/PM_Proiect) Files will be uploaded under namespace

**:pm:prj2025:cc:your\_name**. The archive will contain: \* *src/* — all *.c* and *.h* source files \* *schematics/* — KiCad or draw.io schematic files \* *README.md* — project description and build

instructions \* *Makefile* — compile and flash script (*avr-gcc* + *avrdude*) \* *CHANGELOG.md* — version

history </note> ===== Jurnal ===== <note tip> | Week | Activity | |—|—| | Week 1 | Component

selection, initial schematic, MQ-3 and ATmega324P datasheet study | | Week 2 | Hardware

implementation, prototyping | | Week 3 | Software implementation, fixing wrong read values on MQ-3

sensor | </note> ===== Bibliografie/Resurse ===== <note> **Resurse Hardware:** \* [MQ-3](#)

[Datasheet - Hanwei Electronics](#) \* [ATmega324P Datasheet - Microchip Technology](#) \* [LM35 Datasheet - Texas Instruments](#)

**Resurse Software:** \* [AVR-GCC / AVR-libc documentation](#) \* [Microchip Studio \(AVR Studio\)](#)

\* [Microchip AN2480 - I<sup>2</sup>C/TWI for AVR](#) \* PM Lab materials - course wiki (laboratoare GPIO,

Înteroperari, Timere, ADC, I<sup>2</sup>C) </note> <html><a class="media mediafile mf\_pdf"

[href=""?do=export\\_pdf">Export to PDF</a></html>](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2026/theodor\\_ioan.buliga/mihai.stanciu0107](http://ocw.cs.pub.ro/courses/pm/prj2026/theodor_ioan.buliga/mihai.stanciu0107)



Last update: **2026/05/27 05:53**

