

# Etilotest Digital cu Stocare Cloud

## Introducere

Proiectul constă în realizarea unui Etilotest Digital inteligent, capabil să măsoare concentrația de alcool din aerul expirat și să transmită automat aceste date către o platformă Cloud pentru monitorizare.

- **Ce face:** Sistemul detectează vaporii de etanol folosind senzorul MQ-3, procesează semnalul analogic cu ajutorul unui microcontroller ATmega328P și oferă feedback instantaneu (vizual pe LCD și sonor prin buzzer), trimițând simultan datele prin Wi-Fi către un API extern.
- **Care este scopul lui:** Scopul principal este furnizarea unei metode rapide și accesibile de testare a alcoolemiei, asigurând în același timp o evidență digitală a măsurătorilor prin integrarea IoT.
- **Ideea de la care am pornit:** Am pornit de la dorința mea de a-mi proteja prietenii, creând un dispozitiv care să elimine incertitudinea la volan. Am conceput acest etilotest ca un instrument de prevenție responsabil, menit să reducă riscurile rutiere și să ofere o metodă rapidă de verificare a alcoolemiei înainte de plecarea la drum.
- **De ce este util:** Pentru mine, acest proiect este o oportunitate de a învăța cum să conectez hardware-ul analogic la puterea Cloud-ului. Pentru ceilalți, consider că este un instrument vital care poate salva vieți, oferindu-le prietenilor mei și comunității o metodă simplă de a lua decizii corecte și responsabile înainte de a urca la volan.

## Descriere generală



Modul Hardware	Descriere Tehnică	Interacțiune / Protocol
Arduino UNO R3 (ATmega328p + ATmega16u2)	Unitatea centrală de procesare. Gestionează logica locală și calculele.	Master-ul sistemului; coordonează toate perifericele și trimite date către gateway.
Senzor MQ-3	Senzor chimic pentru detectarea vaporilor de etanol (alcool).	Analog (ADC): Trimite semnal variabil (0-5V) către pinul A0 al ATmega328P.
RPi Pico 2WH	Gateway Wi-Fi bazat pe arhitectură ARM.	UART: Primește date seriale de la ATmega și le transmite în Cloud prin Wi-Fi.
LCD 1602	Ecran pentru afișarea în timp real a rezultatelor.	I2C: Primește comenzi de afișare pe 2 fire (SDA/SCL) de la ATmega328P.
Buzzer Pasiv	Componentă pentru avertizare sonoră.	PWM: Generare tonuri de frecvențe diferite în funcție de concentrație (Pin D9).

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.

Exemplu de schemă bloc: <http://www.robs-projects.com/mp3proj/newplayer.html>

## Hardware Design

### Descriere componente

- [Arduino UNO R3](#)
- [Senzor MQ-3](#)
- [RPI Pico 2WH](#)
- [LCD 1602](#)
- [Buzzer Pasiv](#)
- [Set Jumper Wires](#)
- [Breadboard](#)
- [Set Rezistențe](#)

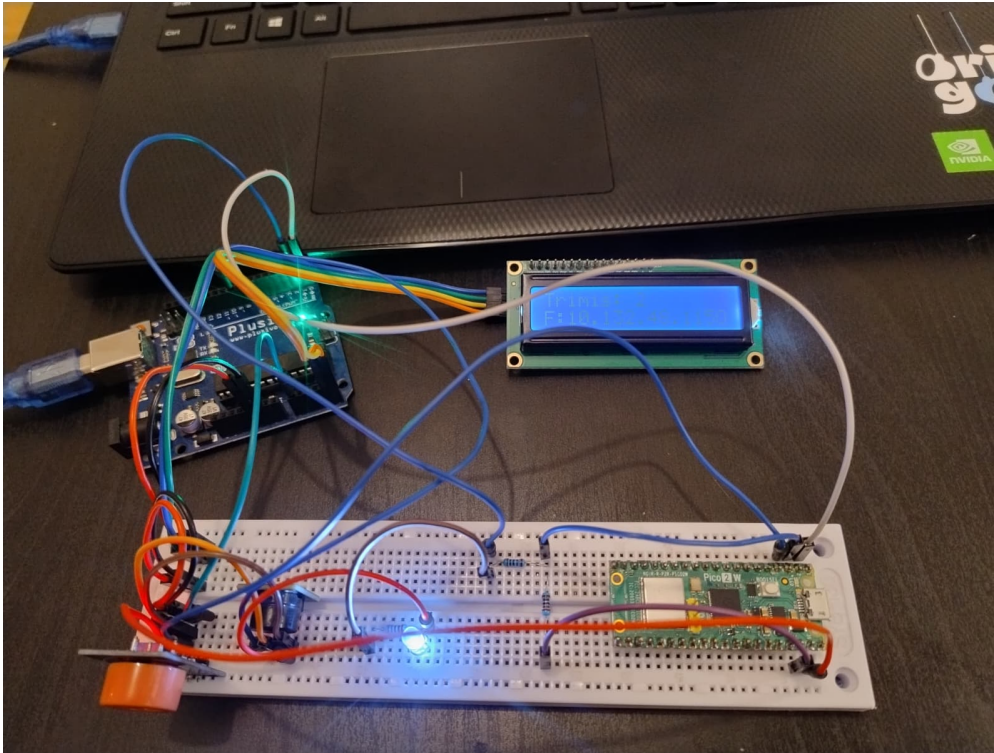
### Tabel Conexiuni Hardware

Componentă	Pin Componentă	Destinație (Placă/Pin)	Detalii
<b>Senzor MQ-3</b>	VCC	Arduino 5V	Necesită preîncălzire pentru precizie
	GND	Arduino GND	
	AO	Arduino A0	Citire concentrație alcool
<b>LCD I2C</b>	VCC	Arduino 5V	
	GND	Arduino GND	
	SDA	Arduino A4	Conexiune Date I2C
	SCL	Arduino A5	Conexiune Ceas I2C
<b>Buzzer Pasiv</b>	VCC	Arduino 5V	
	GND	Arduino GND	
	S	Arduino D9	Generare alerte sonore
<b>LED Debug</b>	Anod (A)	Arduino D13	Serie cu rezistență 220Ω
	Catod (C)	Arduino GND	
<b>Pico 2WH</b>	VSYS	Arduino 5V	Intrare alimentare stabilizată
	GND	Arduino GND	Masa comună
	GP0 (TX)	Arduino D0 (RX)	Transmisie date către Arduino
	GP1 (RX)	Divizor Tensiune	Recepție date protejată (3.3V)
<b>Divizor Tensiune</b>	In (R 1k)	Arduino D1 (TX)	Punct intrare semnal 5V
	Out (1k/2.2k)	Pico GP1 (RX)	Punct ieșire semnal ~3.4V
	GND (R 2.2k)	Pico GND	Masă comună

## Schema electrica



## Rezultate



Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării

## Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)


- algoritmi și structuri pe care plănuiți să le implementați
- (etapa 3) surse și funcții implementate

## 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/cezar.zlatea/daniel.ghindea>



Last update: **2026/05/15 20:01**