

SentinelGuard - Sistem de Alarmă Inteligent Multi-Senzor

Autor: Daniel Brinza

Grupa: 334CB

Asistent proiect: Cezar Zlatea

Introducere

SentinelGuard este un sistem de alarmă inteligent pe care l-am construit în jurul plăcii Arduino Uno. Sistemul monitorizează mai multe tipuri de evenimente relevante pentru securitatea unei camere: mișcare în zona protejată, fum sau gaze inflamabile, zgomote puternice și apropierea unei persoane de panoul alarmei. La detectarea unui eveniment suspect, sistemul reacționează local printr-un buzzer, afișează starea curentă pe un ecran TFT color și transmite mesaje de stare/debug prin UART către calculator.

Scopul proiectului meu este integrarea mai multor periferice și protocoale de comunicație într-o aplicație embedded practică. Alarma nu este doar o demonstrație simplă cu un senzor, ci un sistem cu mai multe stări de funcționare, interfață grafică, evenimente marcate temporal și mai multe surse de intrare.

Ideea proiectului a pornit de la un scenariu real: protejarea unei camere, a unui birou sau a unei zone restrânse cu un dispozitiv compact care poate detecta rapid mișcare, fum/gaz sau activitate neobișnuită. Din punct de vedere didactic, am urmărit să folosesc GPIO, întreruperi externe, timere/PWM, ADC, UART, SPI și I2C în același firmware.

Descriere generală

Am organizat sistemul în jurul microcontrolerului, care citește senzorii, actualizează starea alarmei și controlează interfața cu utilizatorul.

Principalele stări de funcționare sunt:

- DEZARMAT - senzorii pot fi monitorizați, dar sirena nu pornește la evenimente normale.
- ARMAT - alarma urmărește activ mișcarea, fumul/gazul, zgomotele puternice și apropierea de panou.
- PRE_ALARMĂ - a fost detectat un eveniment suspect, iar sistemul așteaptă confirmarea sau expirarea unui timp scurt.
- ALARMĂ - buzzerul este activ, ecranul TFT afișează cauza declanșării, iar evenimentul este transmis prin UART.
- MENIU/SETARI - ecran pentru praguri, mod de armare și starea senzorilor.

Am folosit ecranul TFT ca interfață principală de afișare. Acesta afișează starea curentă a alarmei, ora curentă citită de la modulul RTC, ultimele evenimente și valorile senzorilor.

Schemă bloc



În schema bloc se vede fluxul principal al sistemului: senzorii trimit evenimente și valori către placa Arduino Uno, iar microcontrolerul actualizează interfața TFT, controlează buzzerul și transmite log-uri prin UART. Ecranul TFT este conectat prin adaptare de nivel logic, deoarece placa lucrează la logică de 5V, iar ecranul este tratat ca dispozitiv de 3.3V.

Interacțiunea cu utilizatorul

- Sistemul afișează un dashboard cu ora curentă, starea alarmei și starea senzorilor.
- Când un senzor este declanșat, ecranul evidențiază sursa evenimentului.
- Buzzerul generează semnalizare sonoră la depășirea pragurilor.
- UART prin USB este folosit pentru log-uri, debug și comenzi în timpul dezvoltării și demonstrației.

Concepte de microcontrolere acoperite

Concept	Utilizare în proiect
GPIO	Citirea senzorilor digitali și controlul modulelor simple
Întreruperi externe	Detectarea rapidă a mișcării de la senzorul PIR
Timere / PWM	Modele sonore pentru buzzer și temporizări pentru stările alarmei
ADC	Citirea senzorului MQ-2 și a senzorului de sunet
UART	Log-uri și comenzi prin conexiunea serială USB
SPI	Comunicația cu ecranul TFT
I2C	Comunicația cu modulul RTC DS3231 / DS1307

Design hardware

Listă de componente

Componentă	Cantitate	Rol în proiect	Interfață
Arduino Uno	1	Controler principal	GPIO, ADC, UART, SPI, I2C
Breadboard	1	Realizarea prototipului	-
Fire jumper	multiple	Conectarea componentelor	-
Ecran TFT SPI 2.8 inch, 240×320, ST7789V	1	Interfață grafică pentru stare, alerte și valori senzori	SPI

Divizoare rezistive 330R / 510R	4 perechi	Adaptare de nivel logic 5V → ~3V pentru liniile TFT	GPIO / SPI
Senzor PIR HC-SR501 sau echivalent	1	Detectarea mișcării în zona protejată	Intrare digitală
Modul senzor MQ-2 fum/gaz	1	Detectarea fumului și gazelor inflamabile	Intrare analogică
Modul RTC DS3231 / DS1307 compatibil	1	Păstrarea orei pentru log-uri și afișare	I2C
Baterie CR2032	1	Backup pentru modulul RTC	-
Buzzer pasiv	1	Sirenă / alertă sonoră	PWM / semnal digital
Modul senzor de sunet	1	Detectarea zgomotelor puternice	Intrare analogică
Senzor ultrasonic HC-SR04	1	Detectarea apropierii de panoul alarmei	GPIO + măsurare timp
Buton push-button	1	Oprire locală a sunetului / confirmare alarmă	Intrare digitală

Placa de dezvoltare

În implementarea actuală am folosit placa Arduino Uno. Am ales-o deoarece este compatibilă cu modulele din kit, oferă suficienți pini pentru etapa curentă a prototipului și permite dezvoltare rapidă în PlatformIO.

Funcționalități ale plăcii folosite în proiect:

- interfață USB pentru alimentare și upload firmware;
- UART serial pentru debug;
- SPI hardware pentru ecranul TFT;
- I2C pentru modulul RTC;
- ADC pentru MQ-2 și senzorul de sunet;
- GPIO pentru PIR, HC-SR04, buzzer și buton.

Stadiul actual al implementării hardware

În stadiul actual, am realizat și am testat incremental prototipul hardware SentinelGuard pe placa Arduino Uno. Scopul acestei etape a fost să validez individual componentele și apoi să le integrez treptat într-un montaj unic.

În acest moment, am conectat fizic și am testat următoarele module:

- ecranul TFT SPI 2.8 inch ST7789V;
- senzorul ultrasonic HC-SR04;
- buzzerul pasiv;
- senzorul PIR;
- senzorul MQ-2;
- senzorul de sunet;
- modulul RTC DS3231 / DS1307 compatibil I2C;
- butonul de tip silence / acknowledge.

Am realizat integrarea etapizat, deoarece proiectul conține multe periferice și există constrângeri reale legate de numărul de pini disponibili, nivelele logice și stabilitatea alimentării. În mod special, ecranul TFT a necesitat o atenție suplimentară, deoarece panoul funcționează pe logică de 3.3V, în timp ce microcontrolerul lucrează la 5V.

În urma testelor, am observat că convertorul de nivel logic cu 8 canale disponibil în kit nu a oferit un comportament stabil pentru semnalele SPI ale display-ului. Din acest motiv, în montajul hardware real utilizat în prezentare am adoptat o soluție alternativă: divizoare rezistive pentru liniile SPI de ieșire ale microcontrolerului către TFT. Această soluție s-a dovedit stabilă și a permis funcționarea corectă a ecranului.

Componente folosite și rolul lor în proiect

Componentă	Rol în proiect
Arduino Uno	controler principal folosit pentru prototiparea și integrarea modulelor
TFT SPI 2.8 inch ST7789V	interfață grafică pentru afișarea stării sistemului și a valorilor senzorilor
HC-SR04	măsurarea distanței față de panoul alarmei / detectarea apropierii
Buzzer pasiv	semnalizare acustică la depășirea pragurilor sau în starea de alarmă
PIR HC-SR501	detectarea mișcării în zona monitorizată
MQ-2	detectarea fumului / gazelor inflamabile prin ieșire analogică
Senzor de sunet	detectarea zgomotelor puternice prin ieșire analogică
RTC DS3231 / DS1307 compatibil	furnizarea orei curente pentru afișare și log-uri
Buton „Silence”	confirmarea locală a alarmei sau dezactivarea soneriei

Pinii utilizați în implementarea actuală

Maparea de mai jos reflectă stadiul curent al montajului fizic pe care l-am realizat și al testelor efectuate:

Modul	Semnal	Pin utilizat
PIR	OUT	D2
Buzzer pasiv	SIG	D3
HC-SR04	TRIG	D4
Buton silence	ieșire buton	D5
HC-SR04	ECHO	D8
TFT ST7789V	DC	D9
TFT ST7789V	CS	D10
TFT ST7789V	MOSI / SDI	D11
TFT ST7789V	MISO / SDO	D12 (opțional, în unele teste a rămas neutilizat)
TFT ST7789V	SCK	D13
MQ-2	AO	A0
Senzor de sunet	AO	A1
RTC	SDA	A4
RTC	SCL	A5

Explicația alegerii pinilor

- Pinii D10-D13 au fost rezervați pentru interfața SPI hardware a ecranului TFT, deoarece aceștia oferă cea mai bună compatibilitate și cel mai bun debit pentru actualizarea ecranului.
- HC-SR04 folosește doi pini digitali separați: unul pentru impulsul de trigger și unul pentru măsurarea duratei ecoului.
- Buzzerul a fost conectat la D3, pin potrivit pentru generarea de semnal periodic.
- Senzorii analogici MQ-2 și modulul de sunet au fost conectați pe A0 și A1 pentru a permite citirea unor valori variabile și compararea lor cu praguri software.
- RTC-ul a fost conectat la A4/A5, deoarece aceștia reprezintă magistrala I2C standard pe Arduino Uno.
- Butonul de silence a fost conectat la D5 și configurat software cu rezistență internă de tip INPUT_PULLUP, astfel încât apăsarea butonului produce o tranziție la nivel LOW.

Schema electrică și explicații

Am validat schema de principiu atât în simulator, cât și prin montaj fizic. În simulare am verificat traseele, interacțiunea dintre module și maparea pinilor, iar în montajul real am confirmat funcționalitatea și constrângerile practice.

Un aspect important al implementării a fost conectarea ecranului TFT ST7789V. Deoarece acesta folosește logică de 3.3V, iar placa de control operează la 5V, am coborât liniile SPI de ieșire către ecran de la 5V la aproximativ 3V folosind divizoare rezistive. Pentru fiecare linie SPI critică am folosit următoarea structură:

- rezistență de 330 ohmi între pinul Arduino și nodul de semnal;
- rezistență de 510 ohmi între nodul de semnal și GND;
- nodul intermediar a fost conectat la pinul TFT corespunzător.

Această soluție a fost aplicată pentru:

- SCK;
- MOSI / SDI;
- CS;
- DC.

Pinii de alimentare ai ecranului au fost conectați astfel:

- VCC al modulului TFT la 5V, deoarece placa adaptor a display-ului include regulator onboard;
- GND la masa comună;
- LED și RESET la 3.3V, conform testelor hardware realizate.

Toate modulele folosesc masă comună, iar alimentarea este distribuită prin breadboard.

Simulare conexiuni



Figura de mai sus prezintă schema de conectare pe care am validat-o în simulator. Am folosit simularea pentru verificarea mapării pinilor și pentru testarea logicii de integrare înaintea realizării montajului fizic.

Montaj fizic curent

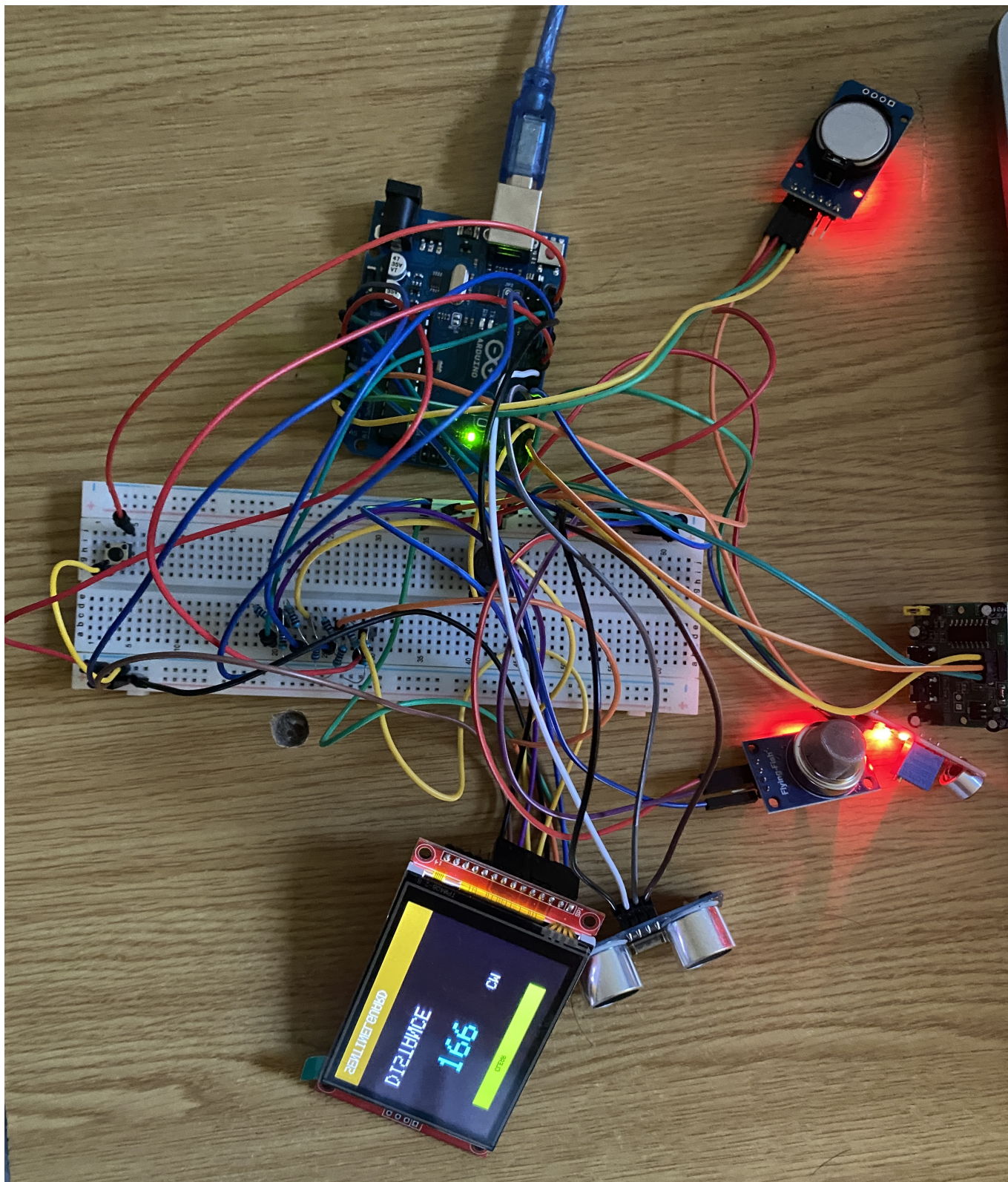


Figura de mai sus prezintă stadiul actual al montajului fizic. Se observă placa de dezvoltare, breadboard-ul central, ecranul TFT conectat prin divizoare rezistive și senzorii principali pe care i-am integrat în jurul sistemului.

Dovadă de funcționare

În stadiul actual am validat funcționarea fiecărui modul hardware principal din proiect. Am verificat individual și în montaj real:

- senzorul ultrasonic HC-SR04;
- senzorul PIR;
- senzorul MQ-2;
- senzorul de sunet;
- buzzerul;
- modulul RTC;
- ecranul TFT.

În imaginea montajului fizic se poate observa afișarea unei valori numerice pe ecran, ceea ce confirmă că interfața de afișare funcționează corect în hardware real. În plus, pe parcursul testelor am confirmat pentru fiecare modul următoarele:

- HC-SR04 transmite distanțe valide, iar valorile pot fi afișate live pe ecran;
- buzzerul reacționează corect la depășirea pragurilor configurate;
- senzorul PIR generează semnal la detectarea mișcării;
- MQ-2 oferă valori analogice care pot fi citite și comparate cu un prag software;
- senzorul de sunet oferă valori analogice utilizabile pentru detectarea zgomotelor puternice;
- modulul RTC poate furniza ora curentă prin magistrala I2C;
- ecranul TFT este inițializat corect și poate afișa valori și stări în timp real.

Prin aceste teste am validat că fiecare senzor și fiecare periferic important din SentinelGuard este funcțional și poate fi integrat în firmware-ul comun al sistemului.

Considerații electrice

- Placa cu microcontroler funcționează la logică de 5V, iar ecranul TFT este tratat ca dispozitiv logic de 3.3V.
- Semnalele SPI și semnalele de control ale ecranului au fost adaptate hardware cu divizoare rezistive.
- MQ-2 are nevoie de o perioadă de încălzire înainte ca valorile citite să devină stabile.
- Buzzerul este comandat de pe un pin digital; dacă în integrarea finală curentul devine insuficient, se poate adăuga un tranzistor driver.
- Toate modulele trebuie să aibă masă comună.

Tipuri de evenimente

Eveniment	Senzor	Reacția sistemului
-----------	--------	--------------------

Mișcare detectată	PIR	Intrare în starea PRE_ALARMĂ sau ALARMĂ
Prag fum/gaz depășit	MQ-2	Intrare imediată în starea ALARMĂ
Zgomot puternic detectat	Senzor de sunet	Intrare în PRE_ALARMĂ, cu prag configurabil
Obiect apropiat de panou	Ultrasonic	Afișare pe ecran și stare de avertizare

Design software

Firmware-ul este scris în C pentru microcontrolerul AVR. Am împărțit implementarea pe etape, astfel încât fiecare bloc hardware să poată fi testat separat înainte de integrarea completă.

Mediu de dezvoltare

- PlatformIO pentru compilarea și încărcarea firmware-ului;
- toolchain AVR pentru Arduino Uno;
- monitor serial pentru log-uri și debug;
- simulator Wokwi pentru verificarea conexiunilor și a logicii hardware.

Module software planificate

Modul	Responsabilitate
sensors	Citește PIR, MQ-2, senzorul de sunet și senzorul ultrasonic
alarm_state	Implementează stările DEZARMAT, ARMAT, PRE_ALARMĂ și ALARMĂ
display_ui	Desenează dashboard-ul TFT și avertizările
rtc_time	Citește data/ora din RTC și formatează timestamp-uri
serial_console	Trimite log-uri prin UART
buzzer	Generează semnalizarea sonoră

Algoritmul principal

```
initializare periferice
afisare ecran de pornire pe TFT
incarcare praguri implicite

while true:
    citire senzori
    actualizare stare alarma
    actualizare buzzer
    actualizare ecran TFT cand apar schimbari
    trimitere evenimente importante prin UART
```

Comenzi UART

Comandă	Efect
STATUS	Afișează starea curentă și valorile senzorilor
LOGS	Afișează ultimele evenimente

Plan de implementare

Am implementat proiectul incremental, astfel încât fiecare componentă hardware să fie testată separat înainte de integrarea finală.

Pas	Activitate	Metodă de validare
1	Alimentarea plăcii și testarea UART	Trimitere și recepție mesaje simple în Serial Monitor
2	Conectarea și testarea buzzerului	Generarea unor beep-uri simple
3	Conectarea senzorului PIR	Afișarea evenimentelor de mișcare prin UART
4	Conectarea senzorului MQ-2	Citirea valorilor ADC și stabilirea unui prag de fum/gaz
5	Conectarea modulului RTC	Citirea datei și orei prin I2C
6	Conectarea ecranului TFT	Afișarea unui ecran de test și a valorilor live
7	Adăugarea senzorului de sunet și a senzorului ultrasonic	Afișarea valorilor și a stărilor pe ecran
8	Implementarea automatului de stări al alarmei	Testarea tranzițiilor între stări
9	Integrare finală și demo	Declanșarea fiecărui senzor și verificarea ecranului, buzzerului și UART

Rezultate așteptate

Rezultatul final așteptat este un prototip funcțional de alarmă multi-senzor. Dispozitivul trebuie să detecteze evenimente de securitate, să afișeze feedback clar pe ecranul TFT, să genereze alerte sonore și să ofere mesaje de debug/stare prin UART.

La finalul proiectului, prototipul trebuie să demonstreze:

- monitorizare în timp real folosind mai mulți senzori;
- interfață grafică clară pe ecranul TFT;
- evenimente marcate cu ora curentă;
- integrarea mai multor protocoale de comunicație într-un singur sistem embedded.

Concluzii

Prin acest proiect am urmărit să combin senzori, protocoale de comunicație și interfață grafică pe un microcontroler cu resurse limitate. Principala provocare a fost integrarea tuturor modulelor astfel încât firmware-ul să rămână responsiv, iar memoria să fie folosită eficient. Ecranul TFT și numărul

mare de senzori cresc dificultatea proiectului, dar fac demonstrația finală mult mai relevantă decât o alarmă bazată pe un singur senzor.

Cod sursă și alte resurse

Codul sursă, schemele electrice, pozele și materialele demo vor fi încărcate într-un repository public de GitHub.

Repository GitHub: <https://github.com/dqnyb/SentinelGuard>

Jurnal

Dată	Progres
09 mai 2026	Alegerea temei: SentinelGuard - sistem de alarmă inteligent multi-senzor. A fost redactată documentația inițială și lista de componente.
13 mai 2026	Am realizat conectarea componentelor hardware pe breadboard și am început testarea lor incrementală.
14 mai 2026	Am adus ecranul TFT în stare funcțională în hardware real, folosind divizoare rezistive pentru adaptarea semnalelor SPI.
15 mai 2026	Am integrat senzorii validați cu afișare live pe ecran și alertă sonoră prin buzzer.
DE_COMPLETAT	Integrarea dashboard-ului final, afișarea simultană a tuturor senzorilor, actualizarea în timp real a valorilor și implementarea logicii complete de alarmă.

Bibliografie / Resurse

- [Pagina proiecte PM 2026](#)
- [Template documentație proiect PM](#)
- [Arduino Uno Rev3](#)
- [Datasheet DS3231](#)
- [Datasheet senzor MQ-2](#)
- [Starter kit](#)
- [Ecran TFT SPI 2.8 inch](#)
- [Modul senzor MQ-2 fum/gaz](#)

[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.brinza>



Last update: **2026/05/15 18:36**

