

# Alarma de Proximitate cu Jurnal Serial

## Introducere

Proiectul constă într-un sistem de securitate inteligent, capabil să monitorizeze distanța și să declanșeze o alertă complexă (vizuală, sonoră și digitală) atunci când este încălcat un perimetru de siguranță configurabil.

- **Ce face:** Măsoară distanța cu HC-SR04, citește pragul dintr-un slide-pot și afișează valori pe OLED SSD1306; la risc, activează LED + buzzer și trimite loguri pe Serial (USART).
- **Scop:** Integrarea interfețelor **ADC, TWI/I2C, Timere/PWM** și **USART** pe ATmega328P.
- **Element de noutate:** Alarmă în doi pași: prealarmă pe distanță, iar alarma sonoră pornește doar la confirmarea mișcării (PIR), cu frecvența buzzer-ului dependentă de distanță.

## Descriere generală

Arhitectura proiectului este centrată pe microcontrolerul **ATmega328P**. Sistemul este organizat în trei piloni funcționali:

### Modulul de Achiziție și Control (Intrare)

- **Senzorul HC-SR04:** Emite ultrasunete și măsoară timpul de î ntoarcere a ecoului pentru a calcula distanța.
- **Senzor Motion (PIR):** Confirmă mișcarea; activează alarma sonoră doar când există intrus real.
- **Slide-pot (ADC0):** Setează pragul de distanță (interval util 5-80 cm).

### Modulul de Procesare

Microcontrolerul compară distanța cu pragul setat și verifică PIR. În funcție de stare, actualizează LED-ul, buzzer-ul și mesajele trimise pe Serial.

### Modulele de Ieșire (Feedback)

- **Feedback Vizual (I2C):** OLED SSD1306 afișează distanța, pragul și starea alarmei.
- **Feedback Sonor și Luminos (PWM):** Buzzer pasiv + LED status; frecvența și ritmul depind de distanță.
- **Jurnal Serial (USART):** Mesaje de stare și alertă către PC (57600 baud).

## Flux de utilizare (pe scurt)

1. Conectează modulele conform tabelului de pini și alimentează placa.
2. Deschide monitorul serial la 57600 baud.
3. OLED-ul afișează distanța și pragul setat de potențiomtru.
4. Reglează pragul (5-80 cm) din slide-pot.
5. În mod OK, LED-ul “respiră” (1s urcare, 1s coborâre).
6. La distanță sub prag: prealarmă (LED constant, log 1/s).
7. La distanță sub prag + PIR activ: alarmă (blink rapid + buzzer, mesaj “!!! INTRUS !!!”).

## Hardware Design

Lista componentelor hardware utilizate:

Nr. Crt.	Componentă	Cantitate	Rol în proiect	Modul / Interfață folosită
1.	ATmega328P-Xplained Mini	1 buc.	Unitatea centrală de procesare.	Toate
2.	OLED SSD1306 128×64	1 buc.	Afișare distanță, prag, stare alarmă.	I2C (SDA, SCL)
3.	Senzor Ultrasonic HC-SR04	1 buc.	Măsurarea distanței prin ultrasunete.	Timere / Input Capture
4.	Senzor Motion (PIR)	1 buc.	Detectarea mișcării.	GPIO
5.	Slide-pot (HW-233)	1 buc.	Reglarea pragului de alarmă.	ADC
6.	Buzzer Pasiv	1 buc.	Alarmă sonoră cu ton variabil.	PWM / Timere
7.	LED status	1 buc.	Alarmă vizuală.	PWM
8.	Rezistență 220Ω	1 buc.	Protecția LED-ului.	N/A

## Pini folosiți și motivație

Componentă	Pin ATmega328P	Rol	Motivație tehnică
OLED SDA	PC4 (SDA)	I2C data	SDA hardware (TWI) pentru I2C stabil
OLED SCL	PC5 (SCL)	I2C clock	SCL hardware (TWI) pentru I2C stabil
Ultrasonic Trig	PD3	output digital	pin simplu pentru impuls 10 us
Ultrasonic Echo	PB0 (ICP1)	input capture	ICP1 măsoară lățimea impulsului precis
Motion (PIR)	PC1	input digital	ieșire digitală comparată intern
Potențiomtru	PC0 (ADC0)	intrare analogică	ADC0 este canal dedicat ADC
Buzzer	PD6 (OC0A)	PWM output	OC0A oferă PWM hardware
LED status	PB3 (OC2A)	PWM output	PWM hardware pentru efecte

## Schema electrică (ASCII)

```

ATmega328P          OLED SSD1306
PC4 (SDA)  ----- SDA
PC5 (SCL)  ----- SCL
VCC        ----- VCC
GND        ----- GND

ATmega328P          HC-SR04
PD3 (TRIG) ----- TRIG
PB0 (ECHO) ----- ECHO
VCC        ----- VCC
GND        ----- GND

ATmega328P          MOTION
PC1 ----- OUT
VCC        ----- VCC
GND        ----- GND

ATmega328P          Slide-pot HW-233
PC0 (ADC0) ----- OUT
VCC        ----- VCC
GND        ----- GND

ATmega328P          Buzzer pasiv
PD6 (OC0A) ----- S
VCC        ----- VCC
GND        ----- GND

ATmega328P          LED status
PB3 (OC2A) ----- ANOD (+)
GND ----- CATOD (-) (cu rezistență serie)

```

## Software Design

### Descrierea codului aplicației (firmware)

Codul este conceput pentru ATmega328P și folosește o arhitectură modulară (driver per periferic). Nu sunt folosite biblioteci Arduino; totul este scris cu acces direct la registre.

- **Mediu de dezvoltare:** VS Code + PlatformIO.
- **Port serial:** 57600 baud pentru jurnal.

## Algoritmi și logică de alarmă

- **Algoritmul de măsurare a distanței:**
  - Se emite un puls de 10μs pe pinul de Trigger.
  - Timer1 (Input Capture) măsoară durata ecoului.
  - Distanța este calculată:  $distanța = (timp\_ecou * 0.034) / 2$ .
- **Procesare ADC (prag):**
  - Valoarea brută (0-1023) este mapată în intervalul 5-80 cm.
- **Stări de alarmă:**
  - **OK:** distanța > prag; LED "respiră" (PWM lent).
  - **Prealarmă:** distanța < prag; LED constant, log 1/s.
  - **Alarmă:** distanța < prag și PIR activ; LED blink rapid + buzzer variabil.
- **USART:** string-uri de status trimise periodic către PC.

## Biblioteci și drivere

Header-ele standard:

- `<avr/io.h>`, `<avr/interrupt.h>`, `<util/delay.h>`, `<avr/pgmspace.h>`
- `<stdint.h>`, `<stdbool.h>`, `<stdio.h>`

Drivere locale:

- `include/usart.h + src/usart.c`
- `include/adc.h + src/adc.c`
- `include/twi.h + src/twi.c`
- `include/ssd1306.h + src/ssd1306.c`
- `include/motion.h + src/motion.c`
- `include/ultrasonic.h + src/ultrasonic.c`
- `include/buzzer.h + src/buzzer.c`

## Structura proiectului și interacțiunea între module

1. Init: USART, ADC, TWI, OLED, motion, ultrasonic, buzzer, GPIO.
2. Buclă: trigger HC-SR04, măsurare distanță, citire prag.
3. Decizie: OK, prealarmă, alarmă (PIR + distanță).
4. Afișare: OLED actualizează valori; serialul loghează starea.

## Logica pe fișiere

- `src/main.cpp`: inițializări, bucla principală, logică de alarmă.
- `src/usart.c + include/usart.h`: inițializare și transmitere USART.
- `src/adc.c + include/adc.h`: citire ADC și mapare prag.
- `src/twi.c + include/twi.h`: comunicație I2C.
- `src/ssd1306.c + include/ssd1306.h`: buffer OLED și randare text.

- `src/ultrasonic.c` + `include/ultrasonic.h`: trigger, input capture, conversie cm.
- `src/motion.c` + `include/motion.h`: citire PIR pe PC1.
- `src/buzzer.c` + `include/buzzer.h`: PWM Timer0 și mapare frecvență.

## Justificarea funcționalităților din laboratoare

- **L1 (USART)**: jurnal serial de alerte.
- **L2 (Timere + întreruperi)**: Input Capture pe Timer1 pentru ecou.
- **L3 (PWM)**: PWM pe OC0A (buzzer) și OC2A (LED).
- **L4 (ADC)**: citire potențiomtru și mapare prag.
- **L6 (TWI/I2C)**: comunicație cu OLED SSD1306.

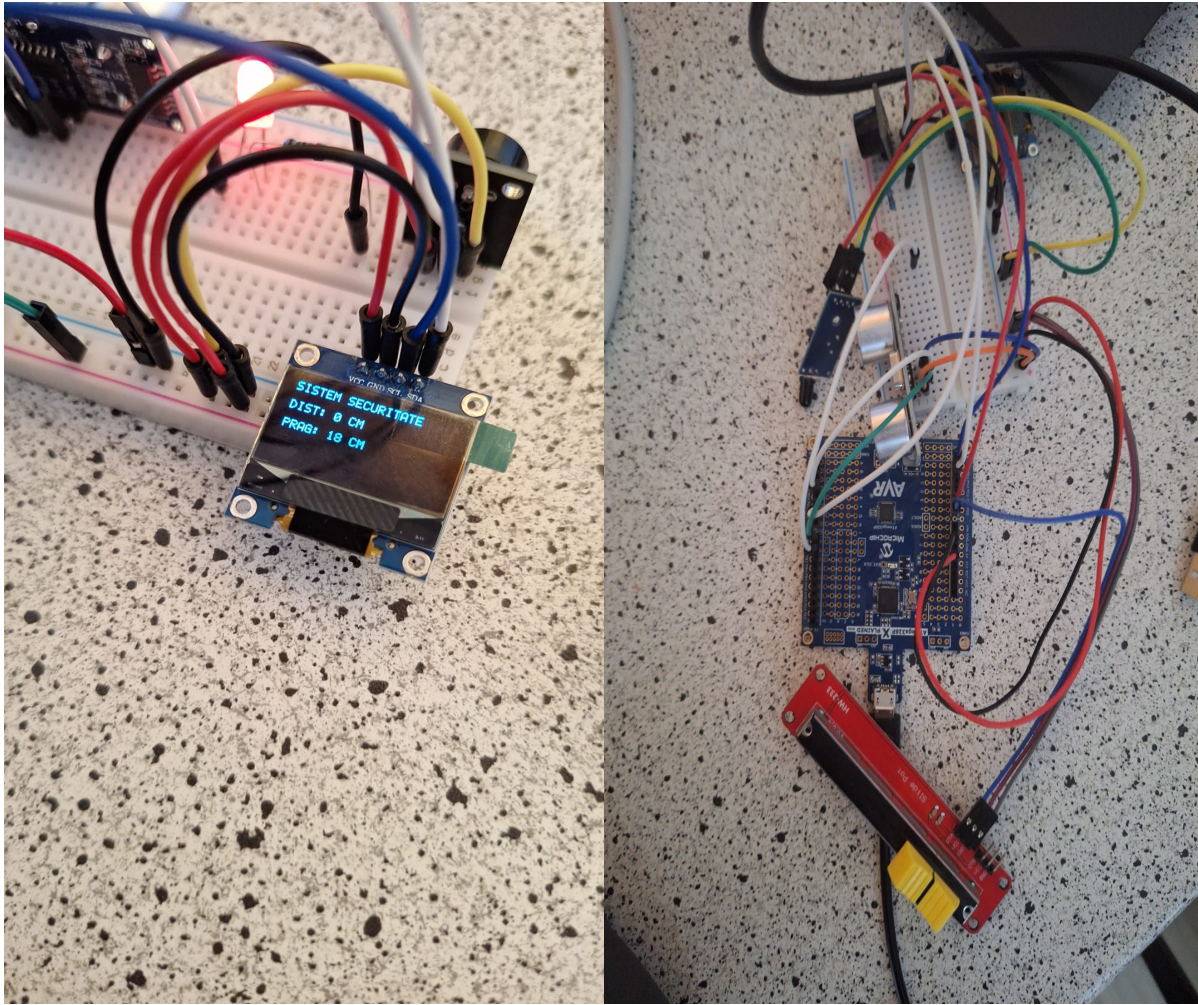
## Observații

- Fontul OLED este minim (cifre, litere mari, câteva simboluri); se poate extinde în `src/ssd1306.c`.

## Rezultate Obținute

Validarea s-a făcut prin:

- monitor serial (mesaje de status și loguri de alertă),
- OLED (distanță, prag, mesaj de intrus),
- LED/buzzer (respirație în OK, blink rapid la alarmă, buzzer activ),
- test practic cu obiect apropiat și mișcare PIR (alarmă confirmată).



## Concluzii

Proiectul demonstrează integrarea corectă a interfețelor ADC, TWI/I2C, Timere/PWM și USART într-un sistem de alarmă funcțional. Soluția oferă feedback clar (vizual, sonor și digital) și permite configurarea pragului în timp real, fiind o bază solidă pentru extinderi viitoare.

## Download

[alarma-proximitate.zip](#)

## Jurnal

- **Milestone 1:** documentație.
- **Milestone 2:** implementare hardware + testare componente.
- **Milestone 3:** implementare software, testare finală.

## Bibliografie/Resurse

- laboratoarele de pe ocw
- documentatia placii ATmega328P si pagina producatorului fiecarui senzor/componente

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/marian.panait1206>



Last update: **2026/05/21 22:24**