

Detector de vibrații în cutremur

Nume: Boari Ioana-Ruxandra

Grupa: 331CD

GitHub Repo: <https://github.com/IoanaBoari/earthquake-vibration-detector>

Introducere

Proiectul reprezintă un sistem de detecție a vibrațiilor, similare celor produse de un cutremur. Atunci când sunt detectate astfel de mișcări, sistemul declanșează alerte sonore, vizuale și afișează un mesaj pe un ecran LCD.

Scopul este de a oferi un semnal de atenționare în timp util, astfel încât utilizatorul să poată interveni pentru a proteja obiectele fragile aflate în apropiere.

Ideea proiectului a pornit de la interesul personal pentru fenomenul cutremurelor și din dorința de a găsi soluții simple pentru a fi cât mai precaută în astfel de situații. Consider util un astfel de dispozitiv deoarece poate oferi un avertisment imediat la apariția vibrațiilor, oferind timp pentru a reacționa și a evita eventuale pagube materiale.

Descriere generală

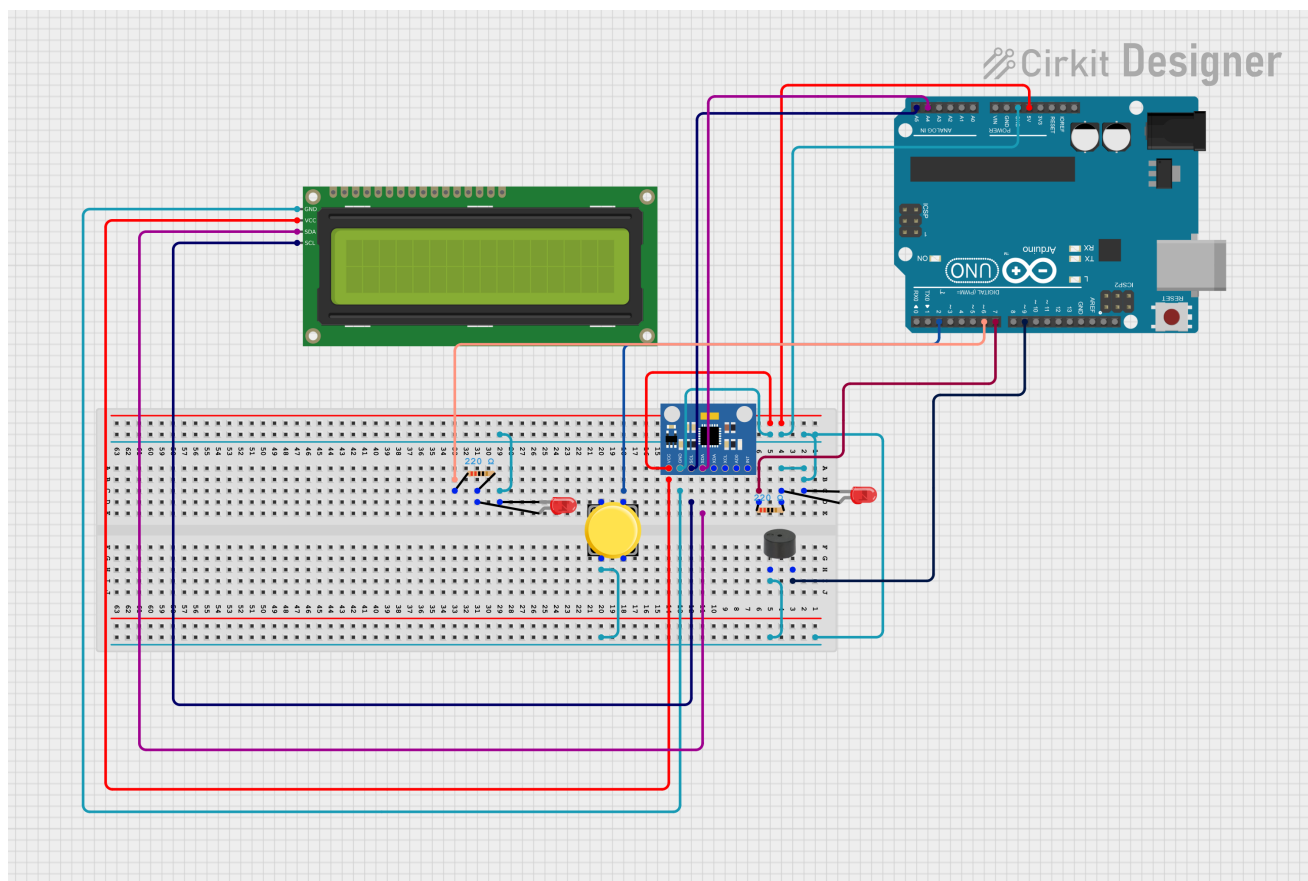
În proiect se utilizează un senzor MPU6050 care transmite către Arduino date de accelerație prin interfața I2C. Arduino compară valorile citite cu valorile de repaus stabilite la calibrare. Dacă diferența depășește un prag predefinit, sistemul consideră că s-a produs o vibrație semnificativă și declanșează o alertă: activează un buzzer controlat prin semnal PWM, aprinde alternativ două LED-uri și afișează un mesaj de avertizare pe un ecran LCD 1602 prin I2C. Alarma va fi resetată printr-un buton conectat la un pin configurat cu întrerupere, iar o nouă apăsare va reactiva sistemul, reluând procesul de calibrare și monitorizare. În plus, sistemul trimite în timp real datele de accelerație prin interfața UART, permițând vizualizarea grafică a acestora în Serial Plotter.



Hardware Design

Listă de piese:

- Schema elettrica:**



Componenta	Procurare	Datasheet
Arduino Uno	Optimus Digital	Datasheet
Accelerometru MPU6050	Emag	Datasheet

LCD 1602	Optimus Digital	Datasheet
Buzzer pasiv	Optimus Digital	Datasheet
LED rosu	Optimus Digital	Datasheet
Buton tactil	Optimus Digital	Datasheet
Rezistentă 220ohm	Optimus Digital	
Breadboard	Optimus Digital	
Cabluri de conectare	Optimus Digital	

Descrierea în detaliu a funcționalității hardware

1. Arduino UNO

- Microcontroller principal care controlează întregul sistem.
- Tensiune de lucru: 5V.
- Interfețe utilizate:
 - **I2C** → pentru comunicarea cu **MPU6050** și **LCD 1602**, folosind pini A4 (SDA) și A5 (SCL).
 - **PWM** → pentru controlul sonorului emis de **buzzerul pasiv** prin funcția `tone()` pe pinul D9.
 - **GPIO** → pentru aprinderea LED-urilor (D6 și D7) și citirea stării butonului (D2).
 - **External Interrupt** → pentru a detecta instantaneu apăsarea butonului (D2), permițând întreruperea execuției normale și comutarea stării sistemului.
 - **UART** → pentru transmiterea datelor de accelerație către **Serial Monitor** și **Serial Plotter**.
- Motivație: Arduino UNO este compatibil cu toate modulele utilizate, are suport extins de biblioteci și este ideal pentru prototipare rapidă, oferind acces facil la toate interfețele necesare proiectului.

2. MPU6050 - Senzor de accelerație și giroscop

- Comunicare: Interfață I2C.
- Pini utilizați:
 - SDA → **A4** (I2C Data)
 - SCL → **A5** (I2C Clock)
 - VCC → **5V**
 - GND → **GND**
- Motivație: I2C permite conectarea mai multor dispozitive pe aceiași doi pini. MPU6050 oferă accelerație pe 3 axe, ideală pentru detecția vibrațiilor.

3. LCD 1602 cu interfață I2C

- Comunicare: I2C (partajează magistrala cu MPU6050).
- Pini utilizați:
 - SDA → **A4**
 - SCL → **A5**
 - VCC → **5V**
 - GND → **GND**
- Motivație: Afișează mesaje despre starea sistemului. Interfața I2C simplifică conectarea cu doar 2 pini.

4. Buzzer pasiv

- Comunicare: Semnal digital generat prin PWM (`tone()`).
- Pin utilizat:

- Semnal → **D9** (PWM-capable pin)
- GND → **GND**
- Motivatie: D9 este un pin compatibil cu `tone()`. PWM permite generarea de tonuri de frecvențe diferite pentru semnal de alertă.

5. LED-uri

- Comunicare: GPIO (ieșire digitală).
- Pini utilizați:
 - LED 1 → **D6** (cu rezistență de 220Ω)
 - LED 2 → **D7** (cu rezistență de 220Ω)
 - GND → **GND**
- Motivatie: Pini digitali liberi ușor de controlat. LED-urile oferă feedback vizual în caz de cutremur.

6. Buton tactil

- Comunicare: GPIO + External Interrupt.
- Pini utilizați:
 - Semnal → **D2** (suportă `attachInterrupt()`)
 - GND → **GND**
- Configurat ca: `INPUT_PULLUP` → logică activă pe LOW.
- Motivatie: Pinul D2 este unul dintre puținele pini de pe Arduino UNO care suportă întreruperi externe (INT0). Permite reacție imediată fără busy waiting.

7. Breadboard + cabluri jumper

- Nu comunică direct cu Arduino, dar asigură interconectarea circuitului.
- Tipuri de fire: mamă-tată, tată-tată.
- Motivatie: prototipare rapidă fără lipire.

Software Design

Mediu de dezvoltare

- **Arduino IDE** (versiunea 2.3.6)
- Microcontroller: **Arduino UNO**

Descrierea codului aplicației (firmware)

Proiectul a fost dezvoltat folosind mediul de dezvoltare **Arduino IDE**, în limbajul **C++**, și utilizează funcționalități precum: comunicație I2C, UART, semnal PWM, GPIO și întreruperi externe.

Sistemul are trei stări:

1. **0 - activ**: sistemul monitorizează vibrațiile.
2. **1 - alertă**: se declanșează alarma prin buzzer, LED-uri și mesaj pe ecranul LCD.

3. **2 - standby**: sistemul este oprit până la reactivare prin apăsarea butonului.

Trecerea între stări se realizează folosind o variabilă globală `systemState` și un flag `buttonPressed` setat printr-o întrerupere externă.

Biblioteci utilizate

- `Wire.h` – pentru comunicația I2C cu senzorul MPU6050 și ecranul LCD.
- `LiquidCrystal\_I2C.h` – pentru controlul LCD-ului 1602 prin I2C.

Algoritmi și structuri implementate

- **Calibrare automată**: la fiecare pornire sau repornire, se preiau 100 de citiri de la senzor și se calculează media pentru fiecare axă:

```
accX_ref = sumX / 100;  
accY_ref = sumY / 100;  
accZ_ref = sumZ / 100;
```

- **Monitorizarea vibrațiilor**: în `loop()`, diferențele dintre valorile actuale și cele calibrate sunt comparate cu un prag:

```
int deltaX = abs(accX - accX_ref);  
if (deltaX > threshold) {  
    // declanșează alerta  
}
```

- **Serial Plotter pentru vizualizare live**:

```
Serial.print(accX);  
Serial.print("\t");  
Serial.print(accY);  
Serial.print("\t");  
Serial.println(accZ);
```

- **Gestionarea stărilor**: folosind `systemState`, sistemul comută între monitorizare, alertă și standby.
- **Gestionarea unui buton cu întrerupere externă**: pentru resetarea sau reactivarea sistemului fără polling:

```
attachInterrupt(digitalPinToInterrupt(buttonPin), onButtonPress,  
FALLING);
```

Funcții implementate

- `setup()` – inițializează pinii, senzorii, LCD-ul, Serial și I2C.
- `loop()` – logica principală bazată pe `systemState`.

- ``readAccel()`` – citește valorile X, Y, Z de la senzor.
- ``calibreazaSistemul()`` – afișează countdown și efectuează calibrarea.
- ``playStartupTone()``, ``playShutdownTone()`` – semnale sonore PWM.
- ``stopAlarm()`` – oprește alarma și LED-urile.
- ``onButtonPress()`` – handler pentru întreruperea butonului.

Validare funcționalități

- Sistemul a fost testat în mai multe scenarii prin lovirea suprafeței de sprijin.
- **Serial Monitor** și **Serial Plotter** oferă confirmare în timp real.
- Comportamentul sistemului corespunde așteptărilor:
 1. Alerta se declanșează la vibrații.
 2. Sistemul poate fi oprit și repornit.
 3. Mesajele sunt afișate corect pe LCD.

Elemente de noutate

- **Vizualizare live în Serial Plotter** a valorilor de accelerare.
- **Calibrare în timp real** în paralel cu afișarea countdown-ului.
- **Semnalizare diversă**: luminoasă, sonoră, textuală.

Calibrarea senzorului

- Se face automat la pornire sau reset.
- În timpul afișării countdown-ului de 5 secunde pe LCD, se fac 100 de citiri:

```
for (int j = 0; j < 20; j++) {  
    readAccel();  
    sumX += accX;  
    delay(50);  
}
```

- Media valorilor se folosește ca punct de referință pentru detecția vibrațiilor ulterioare.

Optimizări realizate

- Folosirea delay-urilor precise în ``calibreazaSistemul()`` pentru preluare distribuită a datelor.
- Evitarea polling-ului prin folosirea unei întreruperi pentru buton.
- Funcții separate pentru fiecare responsabilitate logică - **modularizare**.

Rezultate Obținute

În urma realizării proiectului, am reușit să dezvolt un sistem funcțional de detectare a vibrațiilor cauzate de mișcări bruște, care declanșează o alertă sonoră și vizuală și poate fi controlat printr-un buton.

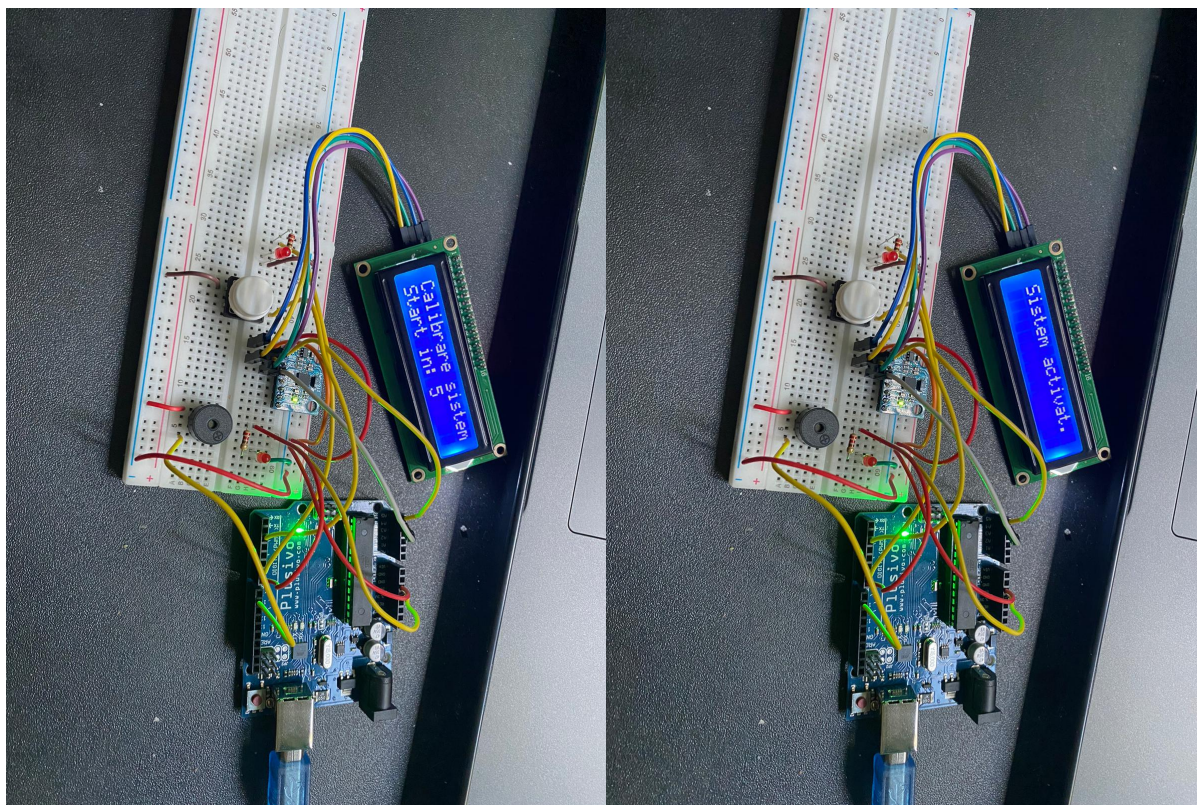
Funcționalități demonstrate

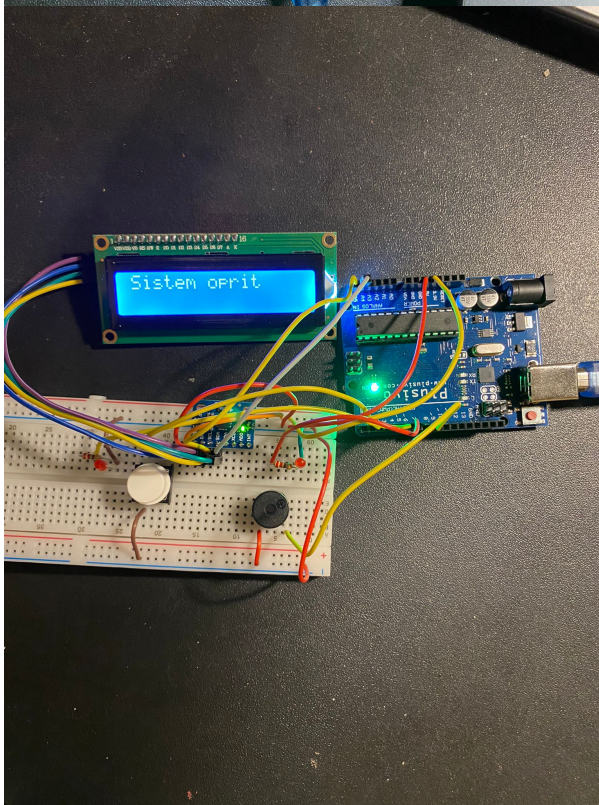
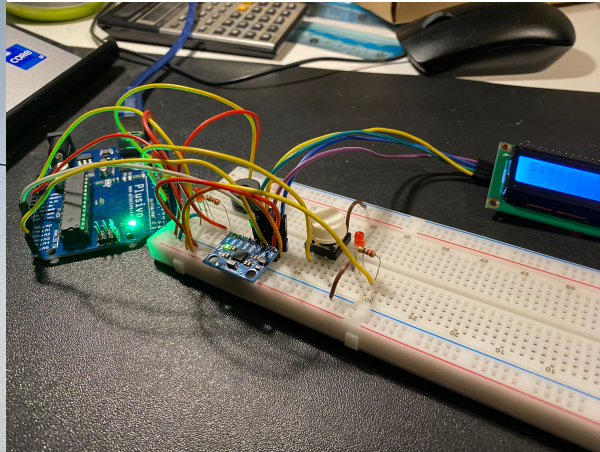
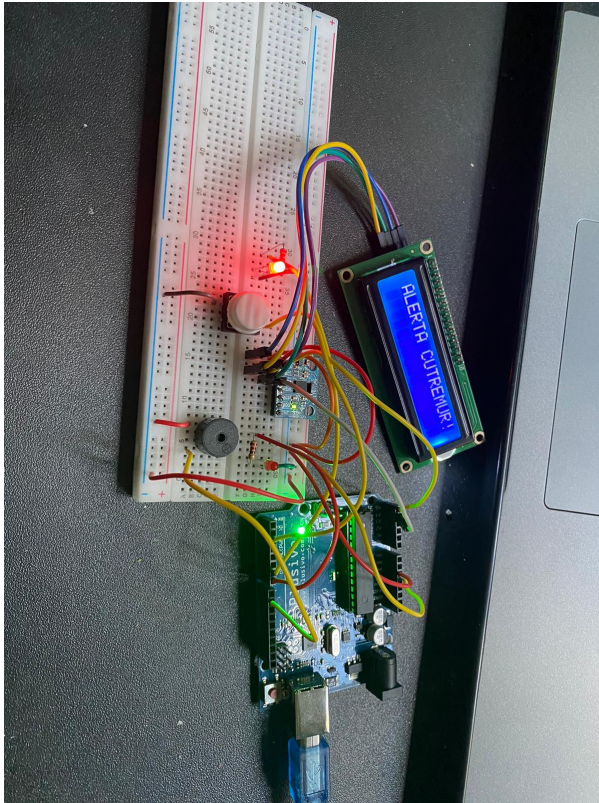
- Sistemul se calibrează automat la pornire
- Detectează variații semnificative de accelerație
- Activează un buzzer pasiv și două LED-uri în mod intermitent
- Afișează mesaj de alertă pe ecranul LCD 1602
- Permite oprirea alertei și trecerea în standby printr-un buton
- Sistemul poate fi reactivat printr-o nouă apăsare

Video demo

https://youtube.com/shorts/2w_z9hhUvBo?si=aUbAqluGFt7ByWH3

Imagini





Concluzii

Sistemul funcționează stabil și a fost testat în mai multe scenarii de simulare a unui cutremur prin lovirea mesei pe care este așezat. Se poate observa clar declanșarea alertei și revenirea la modul standby, confirmând buna integrare a componentelor hardware și logica software.

Mi s-a părut un proiect interesant, care mi-a oferit ocazia să explorez mai multe moduri de a semnala o alertă relevantă pentru oricine, atât vizual, cât și auditiv.

Download

[detector_de_vibratii_in_cutremure.zip](#)

Bibliografie/Resurse

Resurse Hardware

- [Optimus Digital](#) - magazin online pentru componente electronice
- [Modul MPU6050 \(eMAG\)](#) - senzor accelerometru + giroscop folosit în proiect

Resurse Software

- [Arduino IDE](#) - mediu de dezvoltare folosit pentru scrierea și încărcarea codului
- [LiquidCrystal_I2C Library](#) - afișare mesaje pe LCD cu interfață I2C
- [Wire Library](#) - pentru comunicația I2C între Arduino și senzorul MPU6050

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/ioana.boari>

Last update: **2025/05/24 10:19**

