

Tilt Alarm - Pal Roberto-Giulio

1. Introducere

Proiectul propune realizarea unui sistem de securitate inteligent destinat protecției obiectelor de valoare prin monitorizarea miscării și a înclinării. În contextul actual, securitatea fizică a bunurilor necesită sisteme capabile să detecteze nu doar deschiderea unei uși, ci și manipularea sau deplasarea obiectului protejat.

Sistemul utilizează un senzor de mișcare de înaltă precizie (MPU6050) pentru a detecta orice schimbare de unghi sau accelerație bruscă. Proiectul este util deoarece oferă o soluție de securitate compactă, ușor de configurat prin intermediul unei tastaturi matriciale și care oferă feedback în timp real utilizatorului.

2. Descriere generală

Sistemul este construit în jurul plăcii de dezvoltare Arduino Uno. Utilizatorul interacționează cu sistemul printr-o tastatură 4x4 pentru a introduce codul de armare/dezarmare. Odată armat, sistemul preia valorile curente de la accelerometru ca referință. Orice deviație peste un prag stabilit va declanșa o secvență de alarmă.

Funcționalități principale:

Armare/Dezarmare: Securizată prin cod numeric introdus de la tastatură.

Monitorizare activă: Detectarea mișcării pe 2 axe folosind protocolul I2C.

Alertă sonoră și vizuală: Activarea unui buzzer și a LED-urilor în caz de efracție.

Interfață vizuală: Afisarea stării sistemului (ARMED/DISARMED/ALARM) pe un ecran LCD 1602.

Loggig : Transmiterea evenimentelor către un terminal PC prin UART.

Schema de conectare (Diagrama Bloc)



3. Hardware Design

Lista de piese

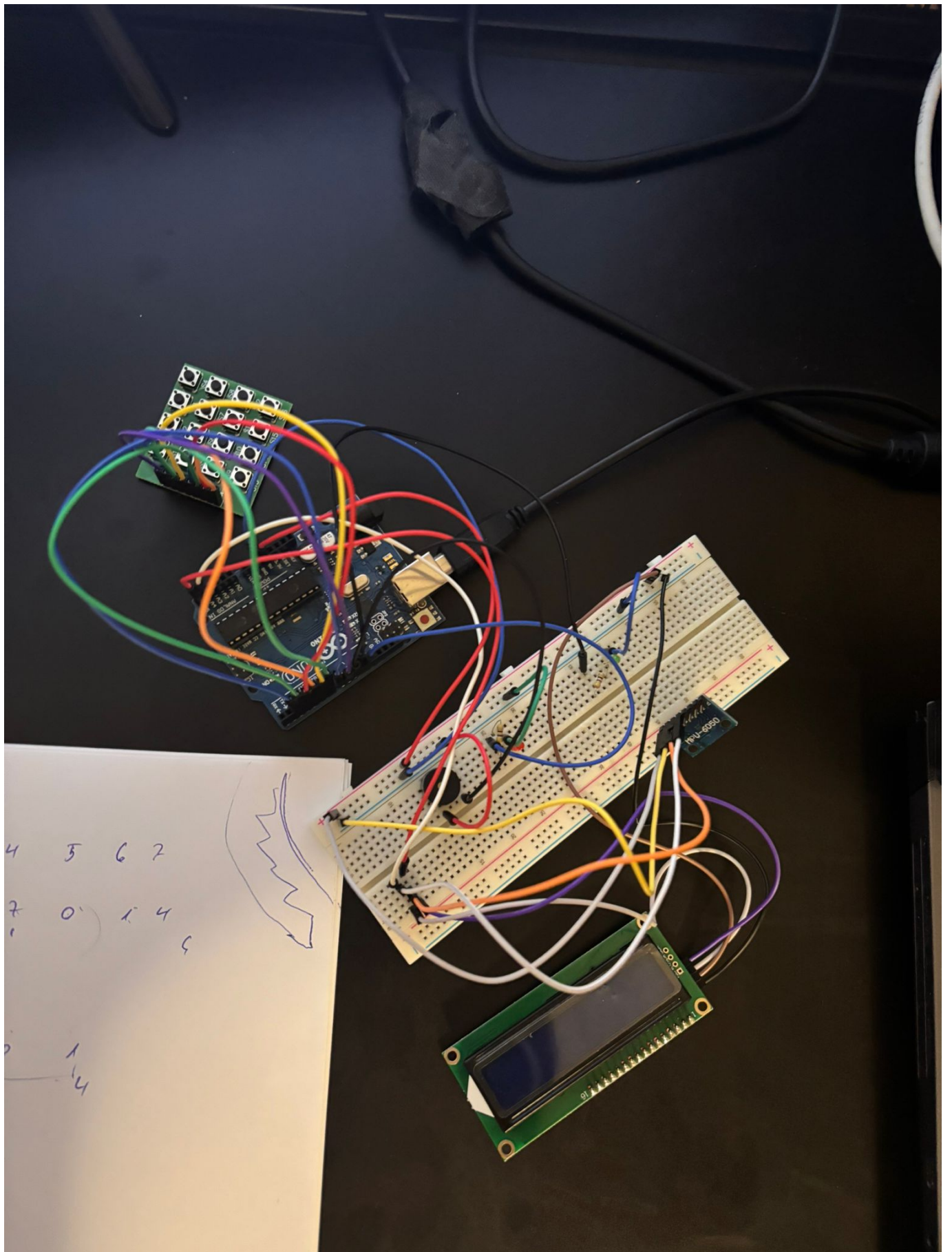
Componenta	Cantitate	Link referinta
Placa de dezvoltare Arduino Uno R3	1	Emag
Modul senzor accelerometru si giroscop MPU6050	1	Optimus Digital
Display LCD 1602 cu modul I2C	1	Emag
Tastatura matriceala 4x4 tip membrana	1	Optimus Digital
Buzzer activ 3V	1	Optimus Digital
LED-uri (Rosu, Verde) + Rezistente 220 Ohm	2	Optimus Digital
Breadboard 830 puncte si fire conexiune	1	Emag

Pinout planificat


Componenta	Pin Arduino Uno	Observatii
LCD 1602 (I2C)	SDA → A4, SCL → A5	Comunicatie I2C (partajata)
MPU6050	SDA → A4, SCL → A5	Adresa I2C diferita de LCD
Tastatura 4x4	D2, D3, D4, D5 (Linii), D6, D7, D8, D9 (Coloane)	Citire prin scanare digitala
Buzzer	D10 (PWM)	Semnal sonor variabil
LED Verde	D11	Indicator stare "Safe"
LED Rosu	D12	Indicator stare "Alarm"
UART	USB / D0 (RX), D1 (TX)	Transmisie date catre PC

Schema Electrica





4 5 6 7
7 0 1 4
5
1 4



4. Software Design

Mediul de dezvoltare utilizat a fost **PlatformIO**.

Codul a fost profilat si gandit pe baza unui **Automat de Stari (State Machine)** non-blocant, avand 3 stari principale: `STATE_IDLE`, `STATE_ARMED` si `STATE_ALARM`. Tranzitiile se fac exclusiv la introducerea corecta a parolei pe tastatura sau la detectarea depasirii pragului de inclinare (`ALARM_THRESHOLD = 10.0` grade). Pentru a asigura responsivitatea sistemului si pentru a evita blocarea citirii tastaturii, functiile blocante (precum `delay()`) au fost complet eliminate din stările active, fiind inlocuite cu un sistem de polling bazat pe `Timer0(millis())`.

Concepte din laboratoare implementate (4 laboratoare):

- **Lab 0 (GPIO):** Aprinderea si stingerea LED-urilor de stare nu se face prin functii standard, ci prin scrierea directa in registrii portului B (`DDRB`, `PORTB`), reducand overhead-ul.
- **Lab 1 (UART):** Sistemul trimite log-uri de stare catre terminalul PC la baud rate 9600. Orice tentativa de efracție sau parola gresita este marcata cu tag-uri de tip `[LOG]` sau `[WARNING]`.
- **Lab 3 (Timere hardware):** Generarea sunetului de alarma a fost optimizata. In loc de functii software consumatoare de timp, a fost configurat **Timerul 1 in modul CTC** (Clear Timer on Compare Match). Prin setarea registrilor `TCCR1A`, `TCCR1B` si `OCR1A`, microcontrolerul genereaza un semnal PWM curat de 1kHz automat in background, direct pe pinul hardware `OC1B (PB2)`.
- **Lab 6 (I2C):** Folosit pentru a interoga senzorul inertial MPU6050 si pentru a trimite date catre ecranul LCD, ambele fiind pe aceeasi magistrala (pinii `A4 - SDA`, `A5 - SCL`), diferite prin adresa I2C.

Biblioteci 3rd Party folosite:

- `Keypad.h` - pentru maparea si citirea matricei 4x4.
- `LiquidCrystal_I2C.h` - pentru controlul display-ului.
- `MPU6050_tockn.h` - pentru extragerea directa a unghiurilor Euler folosind filtrul complementar intern, fara a bloca procesorul cu matematica complexa.

Arhiva proiectului: [Descarca arhiva \(.zip\)](#)

5. Rezultate Obtinute

Sistemul a indeplinit cu succes tintele de performanta stabilite la inceputul proiectului:

- **Timp de reactie si Profilare:** Datorita arhitecturii de tip State Machine cu evaluare bazata pe `millis()` si a delegarii generarii sunetului catre hardware, timpul de executie al functiei `loop()` a ramas sub 50ms. Acest lucru a permis ca la modificarea brusca a unghiului breadboard-ului, alarma sa se declanseze cvasi-instantaneu, fara a rata apasările de taste (0% rata de eroare la debounce-ul tastaturii).
- **Precizia senzorului:** O functionalitate cruciala a fost rutina de auto-calibrare (aprox. 3000 de esantioane calculate la boot). Aceasta ignora pozitia initiala a montajului pe o masa stramba si ia ca referinta (punct de zero) starea exacta din momentul armării.
- **Stabilitate Hardware:** Executia montajului a fost realizata curat, folosind cabluri jumper mama-tata si tata-tata de dimensiuni scurte pentru a minimiza zgomotul electromagnetic pe

magistrala I2C, asigurand lipsa fluctuatiilor la citirea senzorilor.

6. Concluzii

Ipoteza initiala a proiectului a fost confirmata: monitorizarea continua a unui senzor inertial printr-o masina de stari non-blocanta si delegarea proceselor de I/O catre registrii hardware specifici a permis crearea unui sistem de securitate extrem de receptiv.

Elementul de noutate al acestui proiect consta in independenta totala fata de mediul fizic. Majoritatea proiectelor de acest tip folosesc valori prag hardcodate pentru axele X, Y, Z. Prin calcularea offset-urilor giroscopului dinamic, la fiecare alimentare a sistemului, cutia de alarma devine "plug-and-play" pe orice fel de suprafata, indiferent de inclinatia initiala a acesteia.

De asemenea, o alta concluzie trasa in urma dezvoltarii a fost importanta migrarii de la comenzile abstractizate la controlul la nivel de registru. Generarea sunetului hardware pe baza timerelor interne a dovedit ca microcontrolerul poate sustine mai multe actiuni paralele fara a fi incetinit.

7. Bibliografie/Resurse

- Laboratoarele PM - Timere, I2C, GPIO si UART: [OCW PM](#)
- Datasheet ATmega328P: [Microchip.com](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2026/florin.stancu/roberto_giulio.pal



Last update: **2026/05/25 09:30**