

Implementarea unui sistem de alarmă bazat pe  
microcontroler, tastatură, ecran LCD, buzzer, senzori  
de prezență de tip PIR și servomotor  
“passive infrared sensors”

Universitatea Politehnica Timișoara

Betea Adrian Vlad  
Bobu Olimpia-Miruna

# Cuprins

- A. Descrierea proiectului – pag 3
- B. Componente utilizate – pag 3 -8
  - a. Tabel cu componente – pag 3
  - b. Descrierea componentelor – pag 4 -7
  - c. Pini utilizați – pag 8
- C. Arhitectura Hardware – pag 9
- D. Arhitectura Software – pag 10 - 13
  - a. Funcții utilizate – pag 10
  - b. Schemă logică – pag 11
  - c. Diagramă de stare – pag 12
  - d. Biblioteci utilizate – pag 13
- E. Bibliografie – pag 14

## A. Descrierea proiectului

Proiectul constă în implementarea unui sistem de alarmă inteligent și eficient, care utilizează tehnologii moderne și componente electronice pentru a asigura securitatea unei locații. Echipat cu un microcontroler, tastatură, ecran LCD, buzzer, senzori de prezență de tip PIR și un servomotor, acest sistem demonstrează o funcționalitate complexă și o capacitate de răspuns precisă la situații de potențială amenințare.

Funcționalitățile esențiale includ detectarea prezenței neautorizate în încăperi, afișarea mesajelor corespunzătoare pe un ecran LCD cu o rezoluție înaltă și emisia semnalelor sonore distincte, toate acestea fiind gestionate într-un mod inteligent de către microcontrolerul dedicat. Utilizatorul are posibilitatea de a controla sistemul prin intermediul unei tastaturi securizate, permițându-i să armeze și să dezarmeze sistemul folosind un cod special.

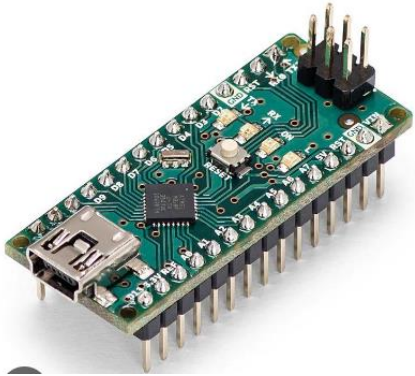
## B. Componente utilizate

### a. Tabel cu componente

Name	Quantity	Component
PIEZ01 PIEZ02	2	Piezo
KEYPAD1	1	Keypad 4x3
PIRPIR1 PIRPIR2	2	PIR Sensor
R1 R2	2	1 kΩ Resistor
SERV01	1	Micro Servo
U1	1	Arduino Uno R3
ULCD	1	PCF8574-based, 32 (0x20) LCD 16 x 2 (I2C)

## b. Descrierea componentelor

### Arduino Nano



Microcontroler. Esența sistemului, fiind responsabil pentru coordonarea tuturor funcțiilor acestuia. Funcționează ca o unitate centrală de control, preluând datele de la diversele senzori și dispozitive, procesându-le și luând decizii corespunzătoare în funcție de algoritmi și programe încărcate în memoria sa. Este practic creierul operațional al sistemului.

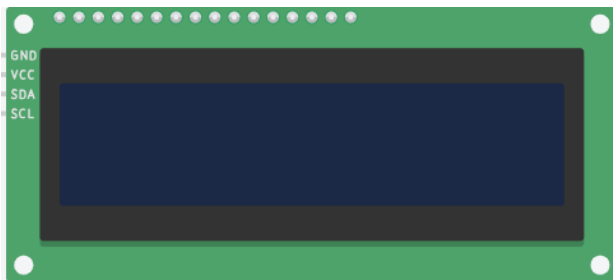
USB connector	
Mini-B USB	
Pins	
Built-in LED Pin	13
Digital I/O Pins	14
Analog input pins	8
PWM pins	6
Communication	
UART	RX/TX
I2C	A4 (SDA), A5 (SCL)
SPI	D11 (COPI), D12 (CIPO), D13 (SCK). Use any GPIO for Chip Select (CS).
Power	
I/O Voltage	5V
Input voltage (nominal)	7-12V
DC Current per I/O Pin	20 mA
Clock speed	
Processor	ATmega328 16 MHz
Memory	
ATmega328P	2KB SRAM, 32KB Flash, 1KB EEPROM
Dimensions	
Weight	7 g
Width	18 mm
Length	45 mm

## 4x3 Keypad



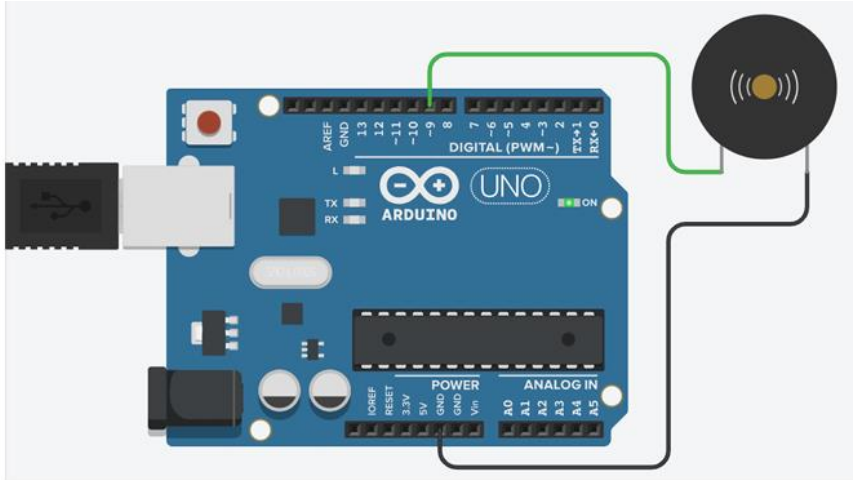
Tastatură. Interfața principală de comunicare între utilizator și sistem. Prin intermediul acesteia, utilizatorul poate introduce comenzi și coduri specifice pentru a controla funcțiile sistemului, cum ar fi armarea și dezarmarea alarmei. Este concepută pentru a asigura un nivel înalt de securitate în procesul de autentificare a utilizatorului.

## I2C LCD Display



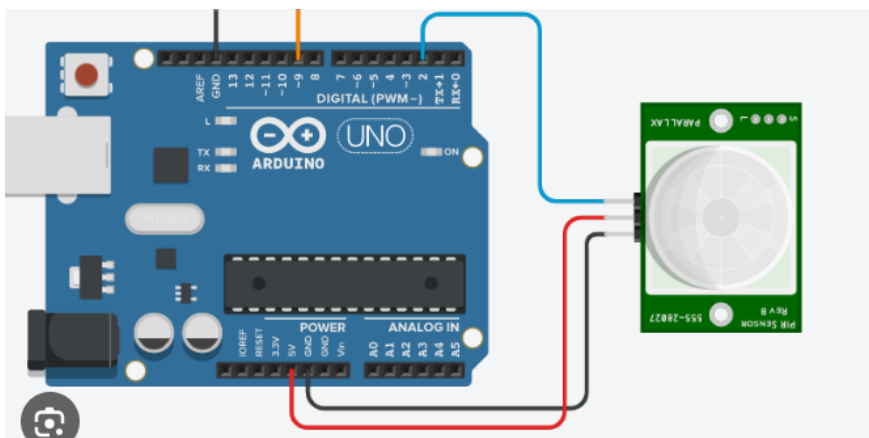
Ecran LCD. Oferă o interfață vizuală pentru afișarea informațiilor și mesajelor relevante utilizatorului. Aici sunt afișate notificările referitoare la starea sistemului, inclusiv avertismente despre prezența neautorizată în încăperi și mesaje de confirmare a operațiunilor efectuate. Este un element esențial pentru comunicarea bidirecțională între sistem și utilizator.

## Buzzer



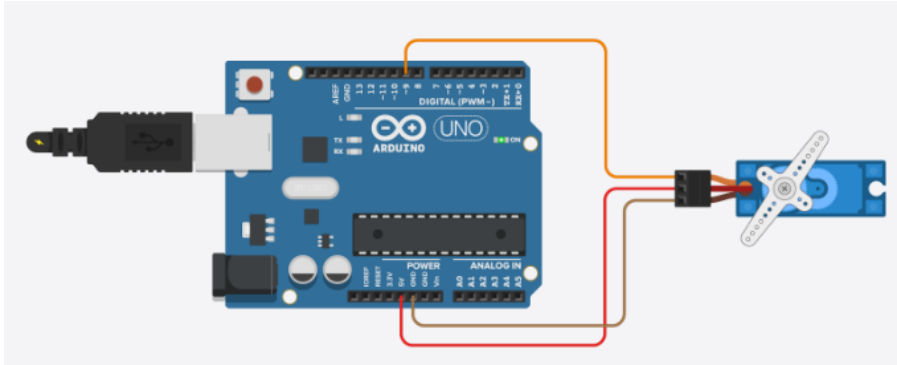
Buzzer: Responsabil pentru generarea semnalelor sonore de alarmă. Atunci când sunt detectate evenimente neașteptate sau prezența neautorizată, acesta emite sunete distinctive pentru a atrage atenția și pentru a avertiza atât utilizatorii cât și posibili infractori despre activarea sistemului de securitate. Este o modalitate eficientă de a transmite informații sonore într-un mod imediat și universal inteligibil.

## PIR Sensors



Senzori de prezență de tip PIR. Utilizați pentru detectarea mișcării în încăperi. Aceștia funcționează prin detectarea schimbărilor în infraroșu generate de corpuri în mișcare în raza lor de acțiune. Astfel, acești senzori sunt esențiali pentru identificarea prezenței umane sau a altor obiecte mobile și activarea alarmei în consecință.

## Servomotor



Servomotor. componentă care simulează acționarea unui sistem de închidere a unei uși. În cadrul acestui proiect, este utilizat pentru a demonstra procesul de închidere a unei uși atunci când sistemul este armat. Este controlat de către microcontroler și poate fi programat să execute anumite acțiuni în funcție de starea sistemului și de comenzile primite.

### c. Pini utilizați

#### SENZORI PIR:

- PIR\_SENSOR1 – pin 2
- PIR\_SENSOR2 – pin 3

#### BUZZERE:

- BUZZER1 – pin 4
- BUZZER2 – pin 5

#### 4x3 KEYPAD:

- ROW1 – pin 12
- ROW2 – pin 11
- ROW3 – pin 10
- ROW4 – pin 9
- COL1 – pin 8
- COL2 – pin 7
- COL3 – pin 6

#### SERVOMOTOR:

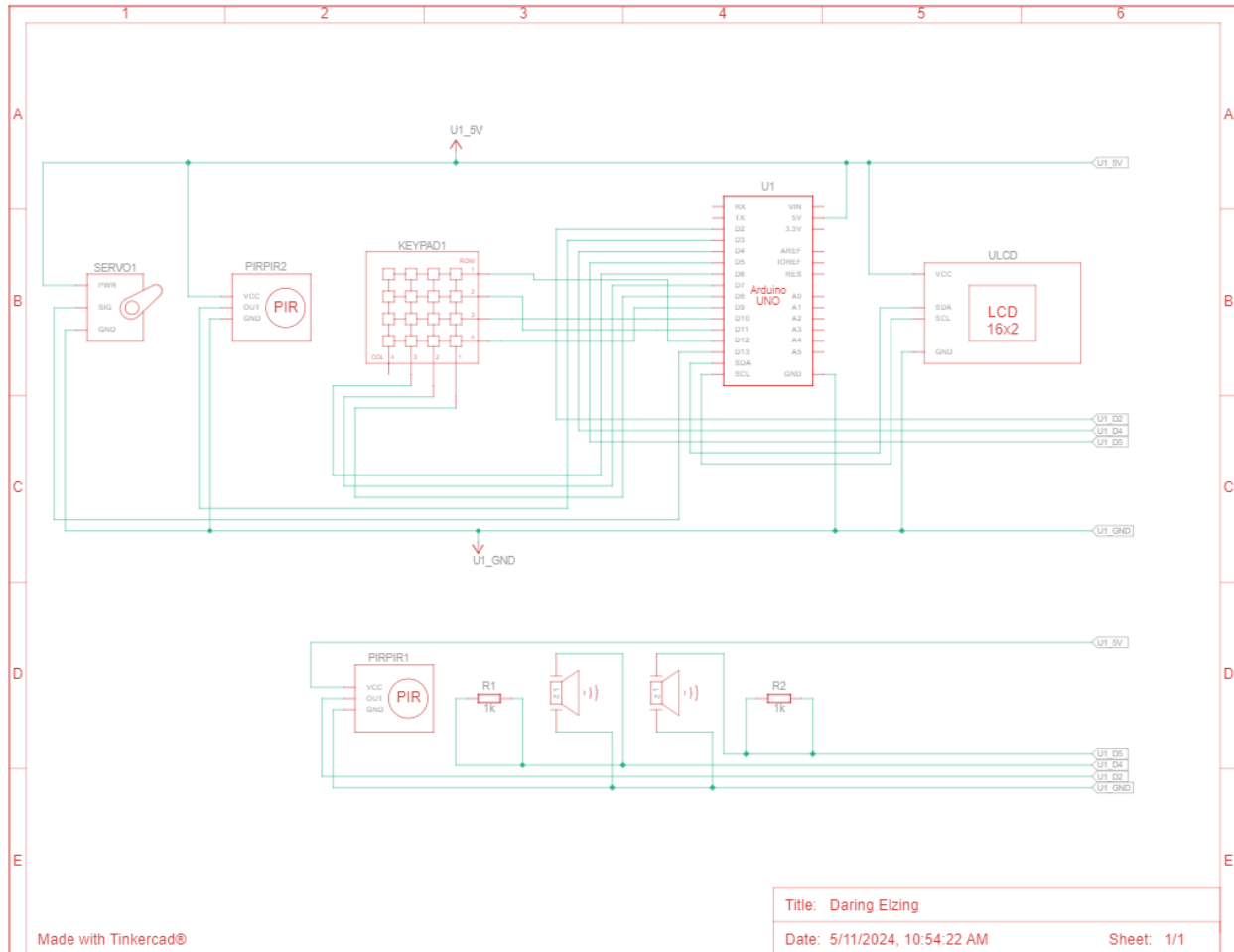
- SERVO\_PIN – pin 13

#### LCD DISPLAY (I2C):

- SDA
- SCL



## C. Arhitectura Hardware



## D. Arhitectura Software

### a. Funcții utilizate

#### ***I. Predefinite***

void setup() - inițializează pinii de intrare și ieșire cât și ecranul LCD

void loop() - repetă constant instrucțiunile din interiorul său

lcd.begin() – funcție care inițializează ecranul LCD

lcd.setCursor() - scriem la o anumită poziție dorită pe ecranul LCD

lcd.print() - afișează date pe un ecran LCD

Serial.begin() - inițializează comunicarea serială între placa Arduino și altă componentă sau computer

digitalRead(COMONENT\_PIN) - citește starea unui PIN digital

millis(); - pentru a evita folosirea repetată a funcției delay()

delay(value) – pune pe pauză programul pentru un număr dat de milisecunde

#### ***II. Implementate de noi***

void alarm() - funcția principală în care avem un meniu folosind switch case

- Case 1 – Alarm disabled

- Case 2 – Citirea codului de armare a alarmei

- Case 3 – Alarma e armată, acționează servomotorul

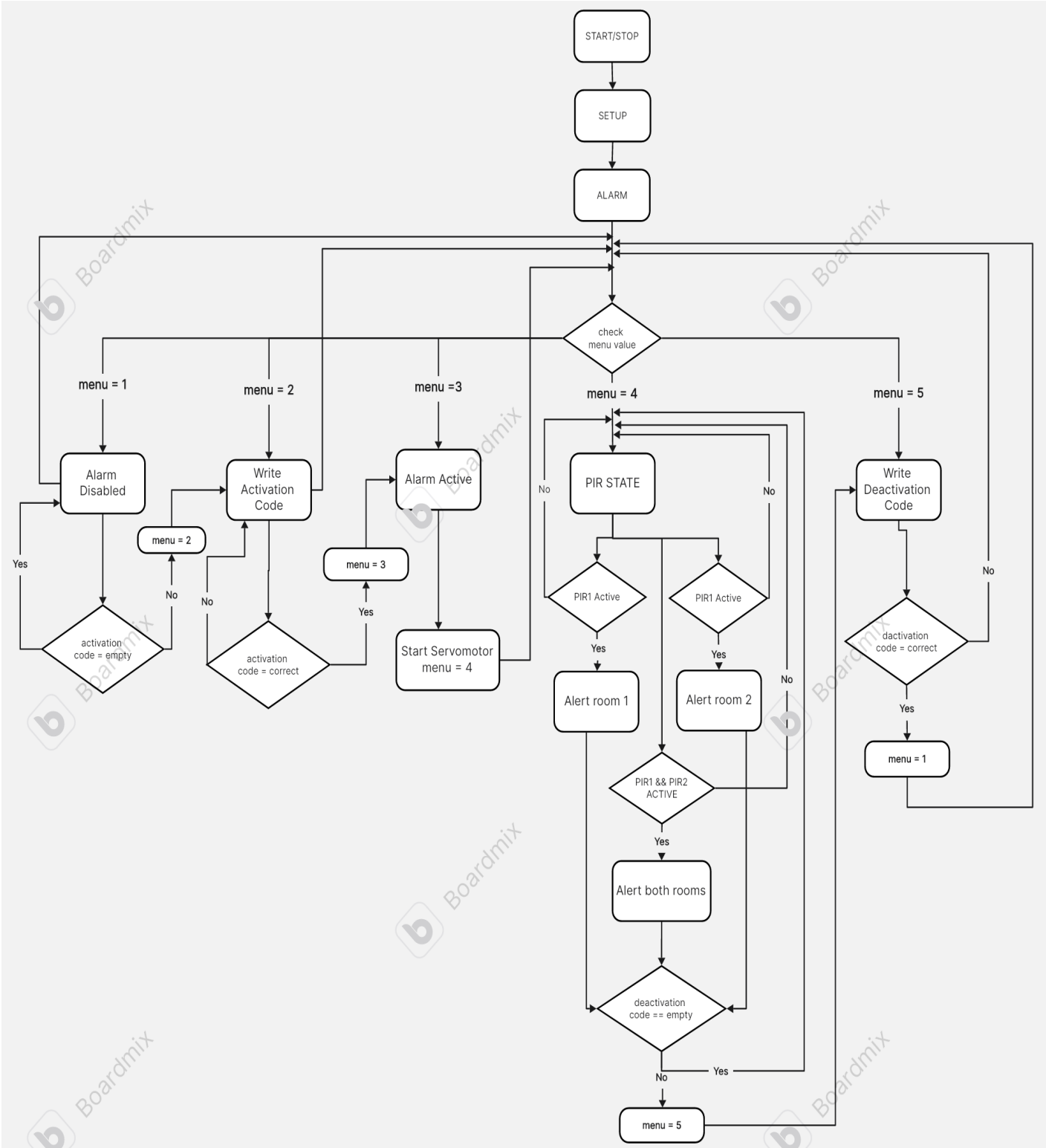
- Case 4 – Apelează funcția pir\_state()

- Case 5 – Citirea codului de dezarmare a alarmei

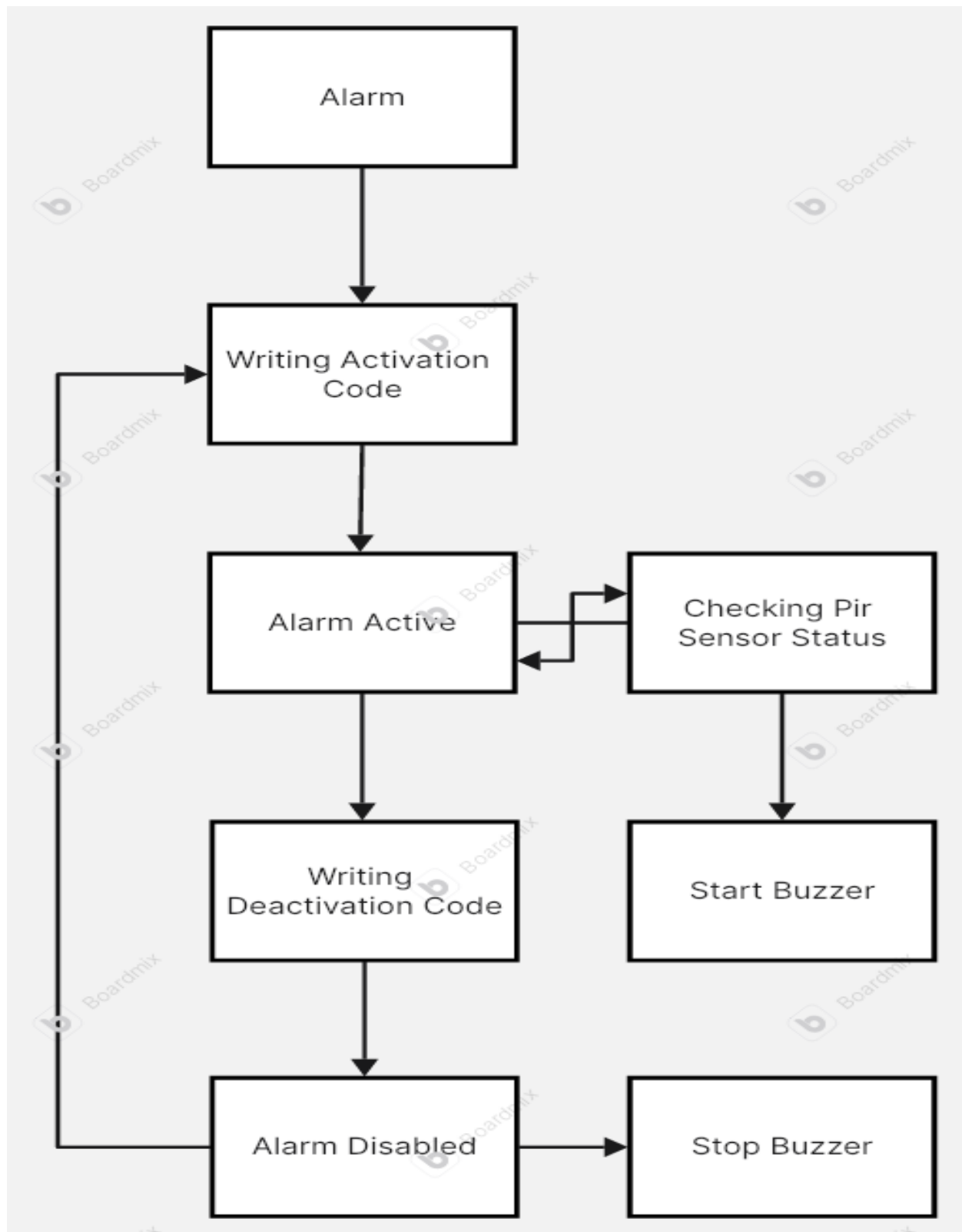
void pir\_state() – funcție ce citește starea senzorilor PIR

void start\_buzzer() – funcție ce activează buzzer-ul, folosește funcția millis() și activează buzzerul pentru un interval dat.

## b. Schemă logică



c. Diagrama de stare



## d. Biblioteci utilizate

### **#include<LiquidCrystal.h> :**

- Folosit pentru a accesa funcțiile necesare pentru a afișa mesaje pe un ecran LCD
- LiquidCrystal\_I2C lcd(0x27, 16, 2); - declararea lcd-ului i2c, situat în zona de memorie 0x27H
- lcd.setCursor(0, 0); – setează coloana și rândul LCD-ului de la care să înceapă mesajul pe care vrem să îl afișăm
- lcd.print("mesaj"); - printează mesajul pe ecranul lcd
- lcd.clear(); - șterge toate caracterele de pe ecranul lcd

### **#include<Servo.h> :**

- Folosit pentru a accesa funcțiile necesare pentru activarea unui servomotor
- Servo servo; - declararea servomotorului
- Servo.attach(SERVO\_PIN)
- Servo.write(0);
- Servo.write(90); - Rotește servomotorul la 90 de grade

### **#include<Keypad.h> :**

- pentru a accesa funcțiile necesare pentru citirea de pe o minitastatură 4x3
- Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
- char key = keypad.getKey(); - salvează în key valoarea citită de la tastatură

### **#include<string.h> :**

- pentru a putea compara, concatena și verifica lungimea unor string-uri

## E. Bibliografie

### Arduino Nano

<https://docs.arduino.cc/hardware/nano/#tech-specs>

### 4x3 Keypad

<https://lastminuteengineers.com/arduino-keypad-tutorial/>

### LCD Display

[https://projecthub.arduino.cc/arduino\\_uno\\_guy/i2c-liquid-crystal-displays-5eb615](https://projecthub.arduino.cc/arduino_uno_guy/i2c-liquid-crystal-displays-5eb615)

### Buzzer

<https://www.arduino.cc/reference/en/libraries/buzzer/>

### PIR Sensor

<https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>

### Servomotor

<https://docs.arduino.cc/learn/electronics/servo-motors/>