

# Smart Home System

## Introducere

**Smart Home System** este un proiect destinat creării unei case inteligente folosind tehnologie accesibilă și integrată. Scopul acestui sistem este de a monitoriza și controla condițiile ambientale, precum și securitatea locuinței, făcând astfel viața mai confortabilă și sigură pentru utilizatori.

Ideea proiectului a pornit de la dorința de a folosi tehnologia pentru a îmbunătăți securitatea și eficiența energetică în locuințe. Prin monitorizarea temperaturii, a umidității, a prezenței gazelor și a mișcărilor nedorite, sistemul poate oferi un răspuns rapid în cazul unei probleme sau poate informa utilizatorul de necesitatea optimizării condițiilor ambientale într-un mod ușor accesibil.

Acest sistem este util atât pentru persoanele interesate de tehnologie, care doresc să își automatizeze locuințele, cât și pentru cei care au nevoie de un sistem robust de monitorizare și control al mediului casnic, crescând astfel confortul și securitatea celor care îl folosesc.

## Descriere generală

### Module Hardware

- **Arduino UNO:** Este responsabil pentru coordonarea tuturor modulelor.
- **Senzori:** Colectează datele despre temperatură, umiditate, prezența gazelor și mișcare.
- **Servomotor:** Acționează ușile/ferestrele pentru ventilare automată în cazul detectării gazelor.
- **Modul de releu:** Controlează becurile, integrând automatizarea iluminatului.
- **Buzzer pasiv:** Emite alerte sonore în cazul detectării mișcărilor nedorite atunci când sistemul de securitate este activ.
- **Ecran LCD:** Afișează informațiile colectate și starea sistemului.

### Module Software

- **Firmware pe Arduino:** Gestionează citirea datelor de la senzori, controlul servomotorului, al buzzerului și al releului. De asemenea, actualizează informațiile afișate pe ecranul LCD.
- **Modul de comunicații:** Permite utilizatorilor să interacționeze cu sistemul prin intermediul unei aplicații mobile.

## Interacțiunea Componentelor

1. Sensorii monitorizează continuu mediul și trimit datele către Arduino UNO.
2. Arduino UNO procesează aceste date, le afișează pe ecranul LCD și, în funcție de parametri setați în procesul de realizare al sistemului, activează servomotorul sau buzzer-ul.
3. Utilizatorii pot comanda manual sistemul de iluminat printr-un modul de comunicații.



## Hardware Design

### Listă de componente:

- 2x Arduino Uno R3 ATmega328P
- 1x Breadboard HQ (830p)
- 1x Breadboard HQ (400p)
- 1x Modul LCD 1602 cu Interfață I2C și Backlight Galben-Verde
- 1x Modul LCD de 2.8" cu SPI și Controller ILI9341 (240x320 px)
- 1x Modul Senzor de Mișcare PIR HC-SR501
- 1x Modul Senzor de Gaz MQ-2
- 1x Senzor de Temperatură și Umiditate DHT11
- 3x Micro Servomotor SG90 90°
- 2x Buzzer Pasiv
- 1x Senzor Ultrasonic HC-SR04
- 4x Condensator 100uF 35V
- 1x Rezistor 10kΩ
- 1x Rezistor 4.7kΩ
- 1x Rezistor 2kΩ
- 1x Rezistor 1kΩ
- 1x Rezistor 220Ω
- 2x Rezistor 100Ω
- 1x Modul Bluetooth HC-05 cu Adaptor
- LED-uri
- Jumper Wires

Din diagrama circuitului din exteriorul casei și din schema electrică a acestuia lipsesc atât Modulul Bluetooth Master Slave HC-05, cât și mai multe LED-uri care vor fi adăugate în circuit atunci când acesta va fi amplasat pe macheta de carton care poate fi observată în imaginea adăugată.

### Diagrama Circuitului din Exteriorul Casei:



## **Schema Electrică a Circuitului din Exteriorul Casei:**



## **Diagrama Circuitului din Interiorul Casei:**

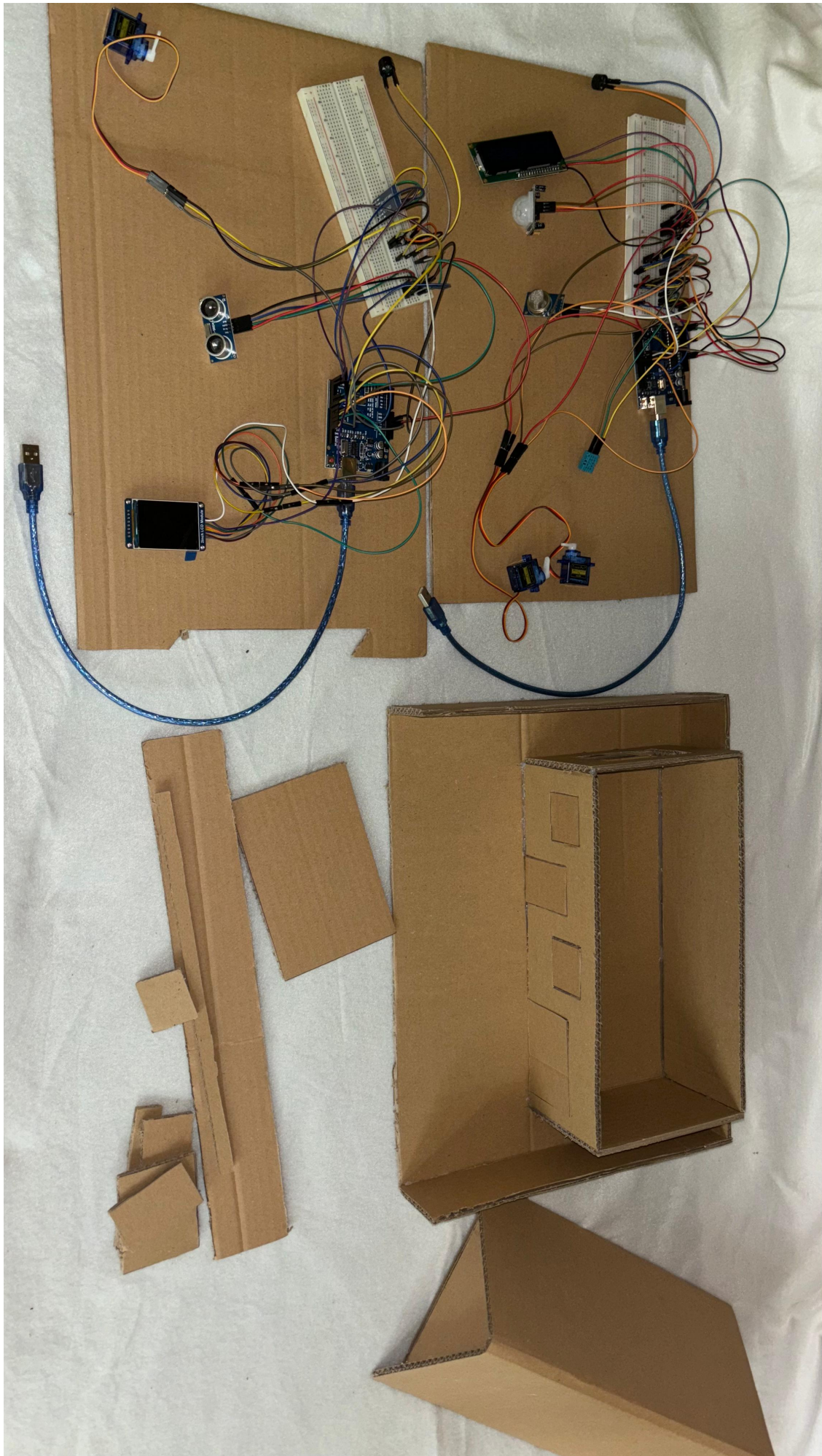


## **Schema Electrică a Circuitului din Interiorul Casei:**



## **Stadiul Actual al părții Hardware:**





# Software Design

## Mediu de Dezvoltare

Codul necesar funcționării proiectului a fost dezvoltat în Arduino IDE în fișierele:

- **circuit\_interior.ino**
- **circuit\_exterior.ino**

Pentru a facilita comunicarea cu modulul Bluetooth HC-05, am folosit **MIT App Inventor**, o platformă web concepută pentru a permite utilizatorilor să creeze aplicații mobile pentru dispozitive Android printr-o interfață simplă de tip drag-and-drop.

## Biblioteci utilizate pentru Circuitul Interior

- **Wire.h** → pentru comunicația I2C între microcontroller și dispozitive periferice, cum ar fi senzori și ecrane LCD.
- **Servo.h** → pentru controlul servomotoarelor, permițând poziționarea precisă a acestora prin semnale PWM.
- **LiquidCrystal\_I2C.h** → pentru controlul ecranelor LCD compatibile cu interfața I2C, facilitând afișarea de texte și date.
- **dht11.h** → pentru utilizarea senzorului de temperatură și umiditate DHT11, oferind metode pentru citirea valorilor de temperatură și umiditate.

## Biblioteci utilizate pentru Circuitul Exterior

- **Adafruit\_GFX.h** → pentru desenarea formelor și textului pe afișaje, oferind funcții esențiale pentru interfețe grafice.
- **Adafruit\_ST7789.h** → pentru controlul ecranelor TFT bazate pe driverul ST7789, facilitând afișarea de imagini și texte.
- **Servo.h** → pentru controlul servomotoarelor, permițând poziționarea precisă a acestora prin semnale PWM.
- **SPI.h** → pentru comunicația serială de mare viteză între microcontroller și dispozitivele periferice folosind protocolul SPI (Serial Peripheral Interface).

## Laboratoare utilizate

- **Laboratorul 1: USART. Debugging** → Utilizat pentru comunicarea serială și debugging.

- **Laboratorul 3: Timere. Pulse Width Modulation (PWM)** → Utilizat pentru controlul precis al servomotoarelor prin PWM.
- **Laboratorul 4: Analog Digital Convertor (ADC)** → Utilizat pentru citirea și procesarea datelor de la senzorii analogici (ex: senzorul de gaz MQ-2).
- **Laboratorul 5: Serial Peripheral Interface (SPI)** → Utilizat pentru comunicația între microcontroller și LCD-ul ST7789 în circuitul exterior.
- **Laboratorul 6: Inter-Integrated Circuit (I2C)** → Utilizat pentru comunicația cu LCD-ul I2C în circuitul interior.

## Descrierea logicii și a metodelor implementate în Circuitul Interior

- **setup()** → folosită pentru inițializarea următoarelor componente:
  - Serial Monitor la 9600 Baud Rate pentru debug.
  - Senzorul PIR setat ca input și LED-ul, buzzerul setate ca output.
  - Butonul setat ca input cu pull-up intern.
  - Servomotoare atașate și setate la poziția inițială.
  - LCD-ul inițializat și setat să afișeze un mesaj inițial.
  - Calibrarea senzorilor timp de 30 secunde.
- **loop()**
  - Verifică starea butonului pentru a comuta între activarea/dezactivarea sistemului de securitate.
  - Monitorizarea senzorului MQ-2 pentru detectarea gazelor și controlul servomotoarelor.
  - Actualizarea afișajului LCD la fiecare 3 secunde pentru a afișa temperatura, umiditatea, valoarea senzorului MQ-2 și starea sistemului de securitate.
  - Monitorizarea senzorului PIR și controlul LED-ului și buzzer-ului.

## Descrierea logicii și a metodelor implementate în Circuitul Exterior

- **setup()** → folosită pentru inițializarea următoarelor componente:
  - Serial Monitor la 9600 Baud Rate pentru debug.
  - Senzorul ultrasonic (TRIG\_PIN ca OUTPUT și ECHO\_PIN ca INPUT).
  - Buzzer-ul, LED-ul setate ca OUTPUT și inițializate la LOW.
  - Servomotorul atașat la pinul specific și setat la poziția inițială.
  - LCD-ul ST7789 inițializat și setat să afișeze mesajele inițiale.
- **loop()**
  - Verifică intrările Bluetooth și actualizează starea LED-ului și poziția servomotorului, respectiv a barierei, în funcție de comenzi.
  - Măsoară distanța folosind senzorul ultrasonic și afișează pe LCD.
  - Actualizează starea de securitate pe baza distanței măsurate.

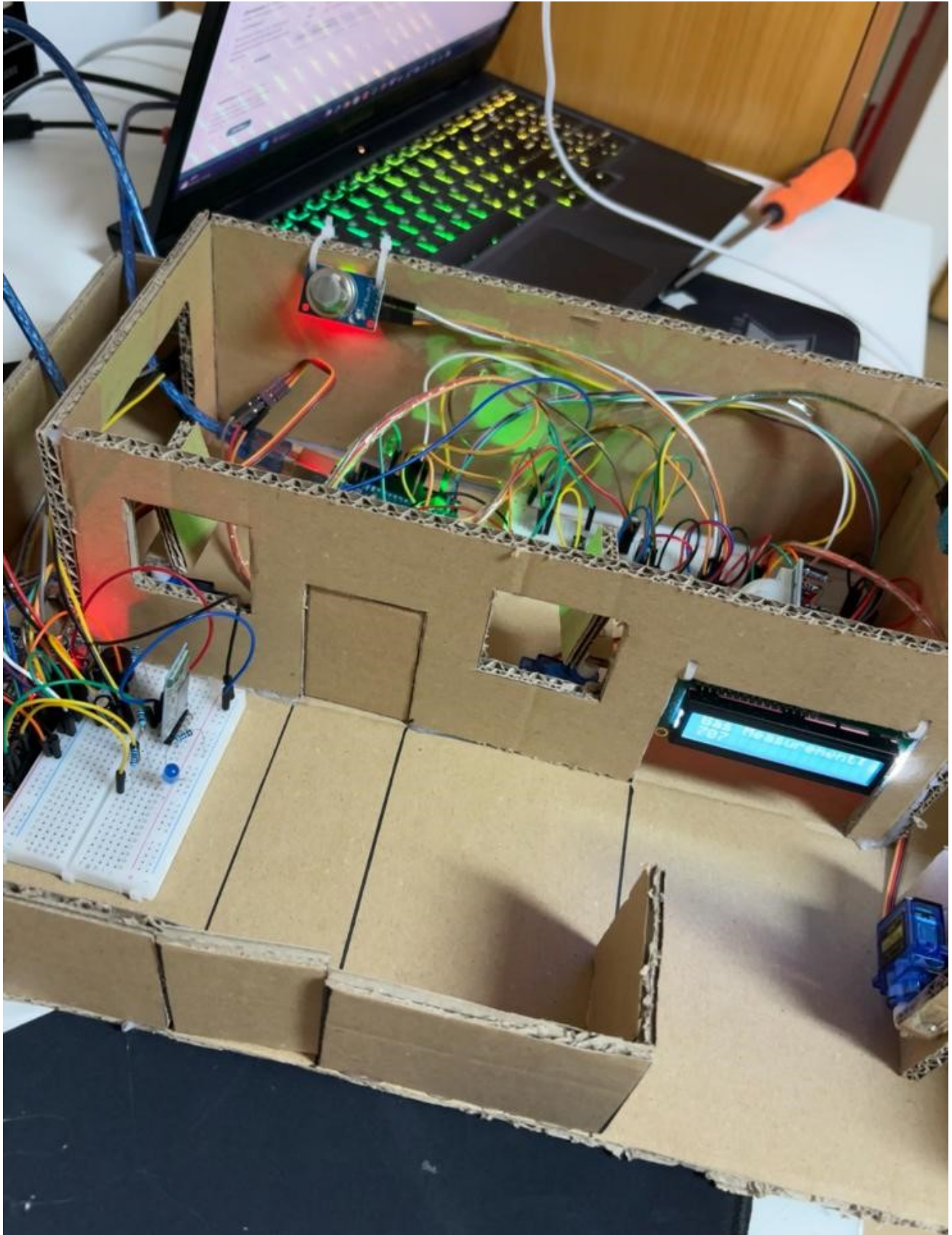
# Descrierea logicii din aplicația de control a Circuitului Exterior



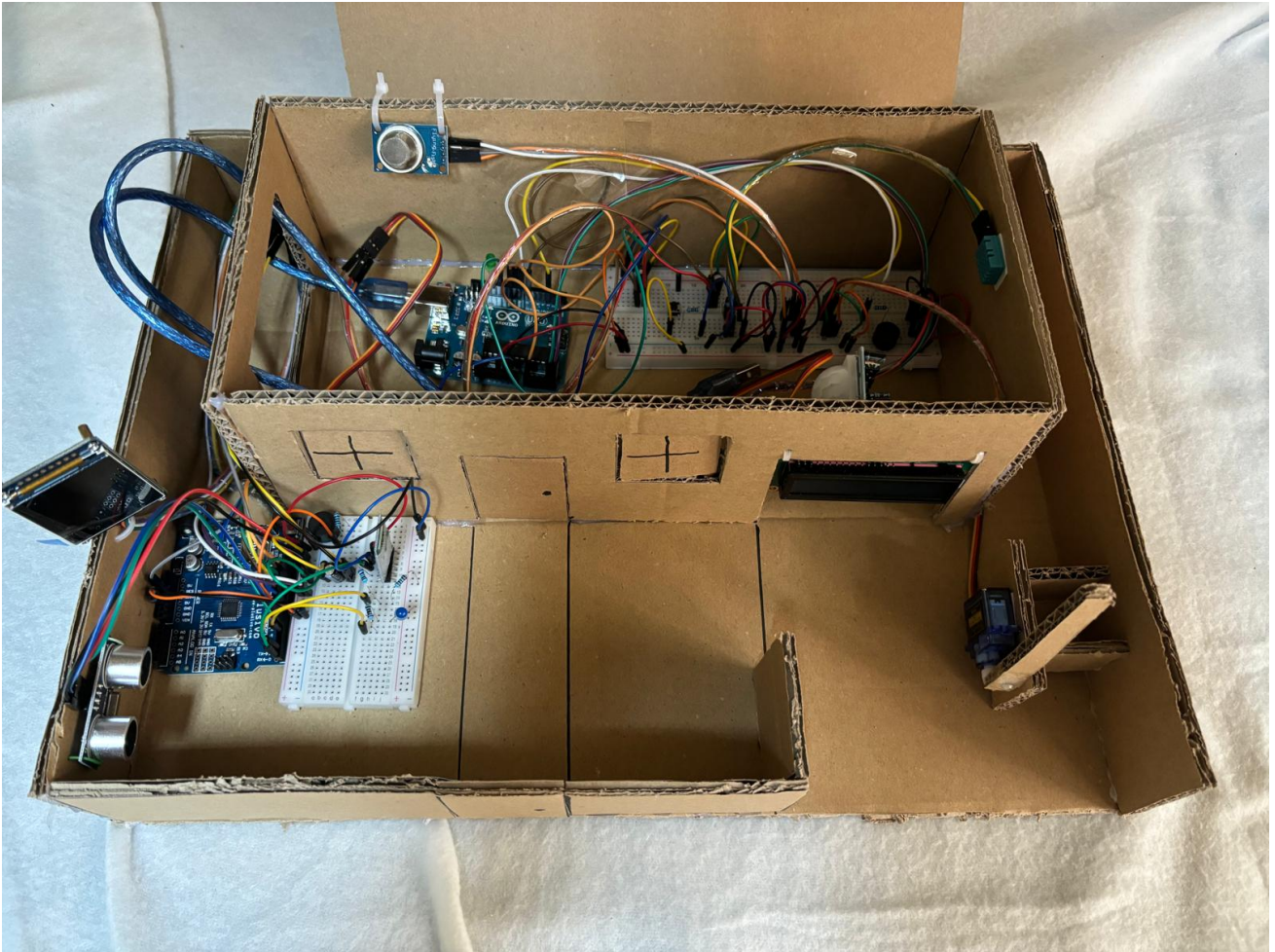
## Rezultate Obținute

## Testul Final al Implementării Proiectului

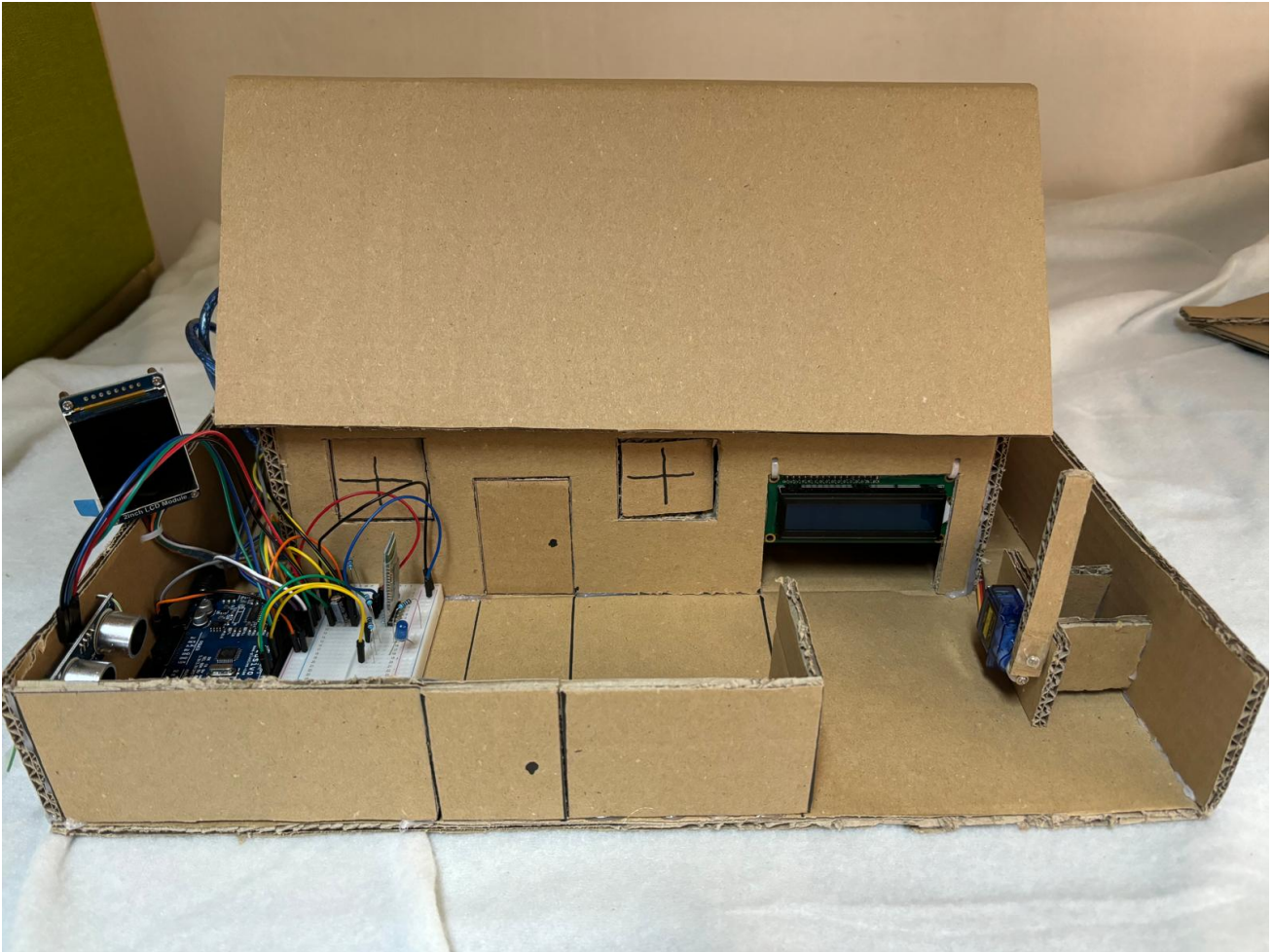
Testul final a constat în verificarea funcționalității tuturor sistemelor integrate atât în circuitul din interiorul casei, cât și în circuitul din exteriorul casei.



## Rezultatul final (1) - Interior



## Rezultatul final (2) - Exterior



## Concluzii

“Smart Home System” reprezintă o metodă de integrare a diferitelor tehnologii pentru a îmbunătăți securitatea și confortul unei locuințe. Utilizând o varietate de senzori și componente electronice, sistemul monitorizează și reacționează la diverse condiții din jurul casei, precum și la cerințele celui care îl folosește.

Implementarea oferă un sistem de securitate cuprinzător atât prin intermediul senzorilor care pot detecta mișcări nedorite în cadrul proprietății, cât și prin intermediul senzorului care activează sistemul automat de aerisire a casei. Integrarea ecranelor LCD permite afișarea în timp real a informațiilor relevante, facilitând utilizatorului monitorizarea constantă a stării sistemului. Controlul servomotoarelor și al altor dispozitive prin intermediul semnalelor PWM și al comunicației Bluetooth adaugă un nivel suplimentar de funcționalitate și flexibilitate. De asemenea, sistemul demonstrează eficiența comunicării seriale pentru debug și control.

În concluzie, acest proiect ilustrează cum tehnologia poate fi folosită pentru a crea soluții interconectate pentru securitatea și gestionarea unei locuințe, oferind o bază solidă pentru extinderea și îmbunătățirea viitoarelor sisteme de automatizare casnică.

## Download & GitHub Repository

Proiectul care constă în aplicație pentru dispozitive Android, cod sursă și alte resurse grafice sunt disponibile pe [GitHub](#).

De asemenea, puteți descărca direct întreg conținutul proiectului accesând: [smart-home-system.zip](#)

## Bibliografie/Resurse

### Resurse Software

- <https://docs.arduino.cc/learn/microcontrollers/analog-input/>
- <https://docs.arduino.cc/learn/microcontrollers/analog-output/>
- <https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>
- <https://www.arduino.cc/reference/en/libraries/dht11/>
- <https://www.arduino.cc/reference/en/language/functions/communication/spi/>
- <https://www.arduino.cc/reference/en/libraries/adafruit-st7735-and-st7789-library/>
- <https://www.arduino.cc/reference/en/libraries/adafruit-gfx-library/>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab0-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab1-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab2-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023-2024>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab4-2023-2024>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab5-2023-2024>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2023-2024>
- <https://ai2.appinventor.mit.edu/>

### Resurse Hardware

- <https://docs.arduino.cc/learn/electronics/servo-motors/>
- <https://docs.arduino.cc/learn/electronics/lcd-displays/>
- <https://support.arduino.cc/hc/en-us/articles/4403050020114-Troubleshooting-PIR-Sensor-and-sensitivity-adjustment>
- <https://newbiely.com/tutorials/arduino-nano/arduino-nano-gas-sensor>
- <https://docs.arduino.cc/learn/communication/bluetooth/>
- <https://docs.arduino.cc/learn/electronics/servo-motors/>
- [https://www.waveshare.com/wiki/2.4inch\\_LCD\\_Module](https://www.waveshare.com/wiki/2.4inch_LCD_Module)
- <https://maxbotix.com/blogs/blog/how-ultrasonic-sensors-work>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/sseverin/robert.nastase0609>



Last update: **2024/05/27 14:28**