

Smart Security System

Introducere

Autor: Vulpe Cezar-Andrei

Grupa: 335CA

Proiectul urmărește realizarea unui sistem de securitate inteligent, capabil să monitorizeze mișcarea în spații interioare și să reacționeze automat atunci când sunt detectate condiții suspecte. Sistemul are rolul de a alerta utilizatorul printr-o notificare și de a emite o alarmă sonoră în momentul declanșării. De asemenea, păstrează o evidență a evenimentelor detectate, care poate fi consultată ulterior. Informațiile legate de starea sistemului și alertele generate sunt afișate local, iar utilizatorul poate interacționa cu sistemul printr-un control simplu și intuitiv.

Motivația din spatele proiectului a fost dorința de a crea o soluție de supraveghere discretă, care să nu invadeze intimitatea locuinței, așa cum o pot face camerele video. Ideea a apărut din nevoia de a avea un mecanism de protecție activ în perioadele în care utilizatorul nu se află acasă, fără a transforma spațiul într-unul supravegheat permanent. Proiectul oferă astfel o alternativă practică, accesibilă și adaptată realității actuale, în care prezența online poate indica absența fizică din locuință.

Descriere generală

Sistemul de securitate este construit în jurul unui microcontroller ESP32 și are ca scop detectarea mișcării în condiții de iluminare scăzută. Acesta integrează un senzor PIR pentru detecția prezenței și un fotorezistor conectat la un pin ADC pentru a evalua nivelul de lumină ambientală. În momentul în care este detectată mișcare pe întuneric, se activează o alarmă sonoră folosind un buzzer controlat prin semnal PWM. Funcționarea alarmei este gestionată de un sistem pe bază de timere, care determină perioadele de activare și pauză, repetate ciclic până la intervenția utilizatorului printr-un buton dedicat. Evenimentele sunt transmise prin Bluetooth sau UART către o interfață de monitorizare externă și sunt logate local pe un card SD prin interfața SPI, fiecare înregistrare fiind marcată cu data și ora obținută de la un modul RTC. Sistemul dispune de un ecran LCD conectat prin I2C pentru afișarea stării curente și a ultimelor alarme, iar întreruperile sunt utilizate atât pentru declanșarea alarmei în baza senzorului PIR, cât și pentru interacțiunea utilizatorului prin butonul fizic.



Hardware Design

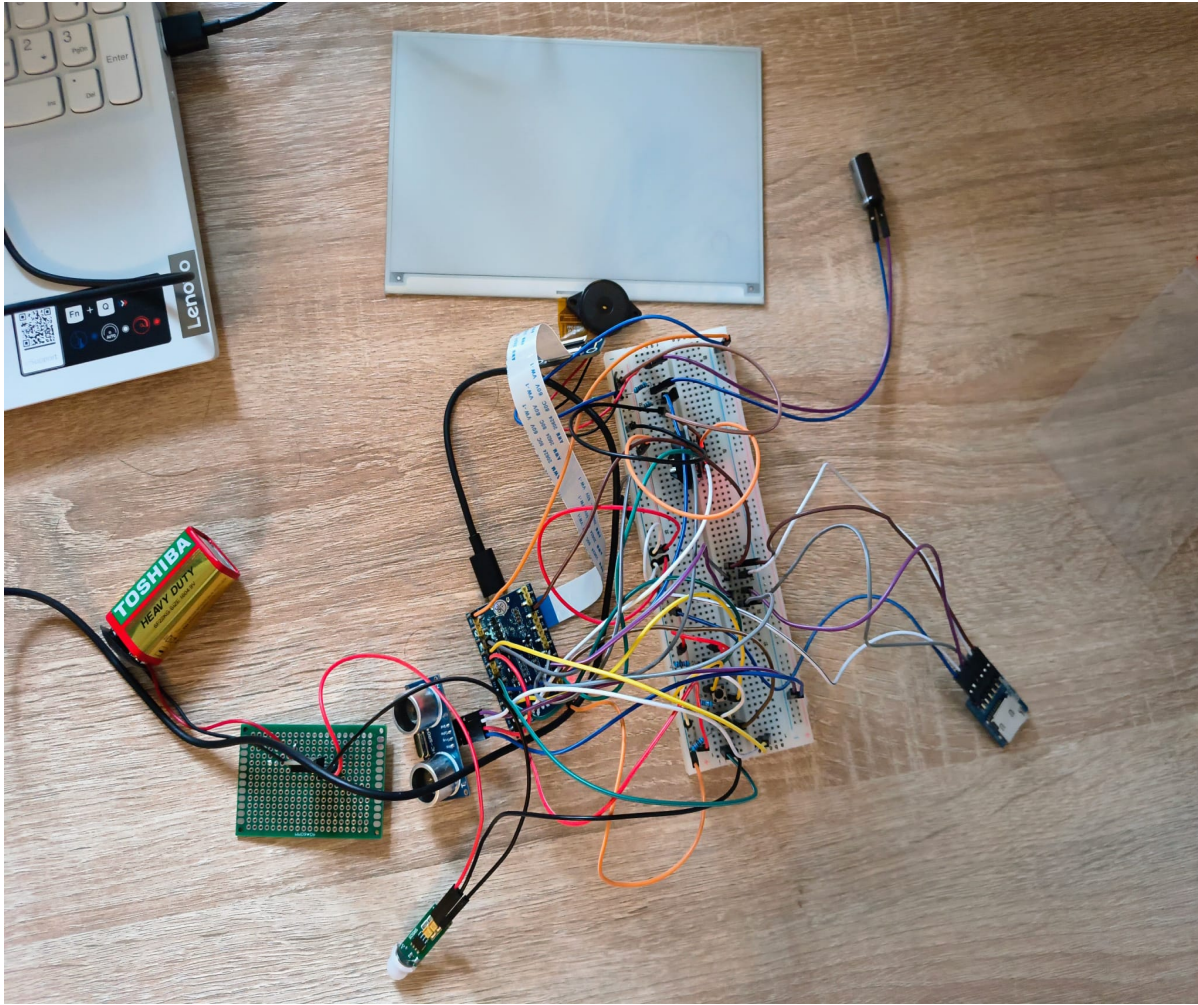
Componente

- [ESP32](#)
- [Senzor miscare\(PIR\)](#)
- [Senzor ultrasonic](#)
- [Ecran EPAPER pe SPI](#)
- [Modul Card SD](#)
- [Modul RTC](#)
- [Convertor tensiune la 5V](#)
- Buzzer
- Fotorezistor
- Baterie 9V
- Butoane
- Rezistente
- Condensator 1mF
- [Tranzistor NPN](#)

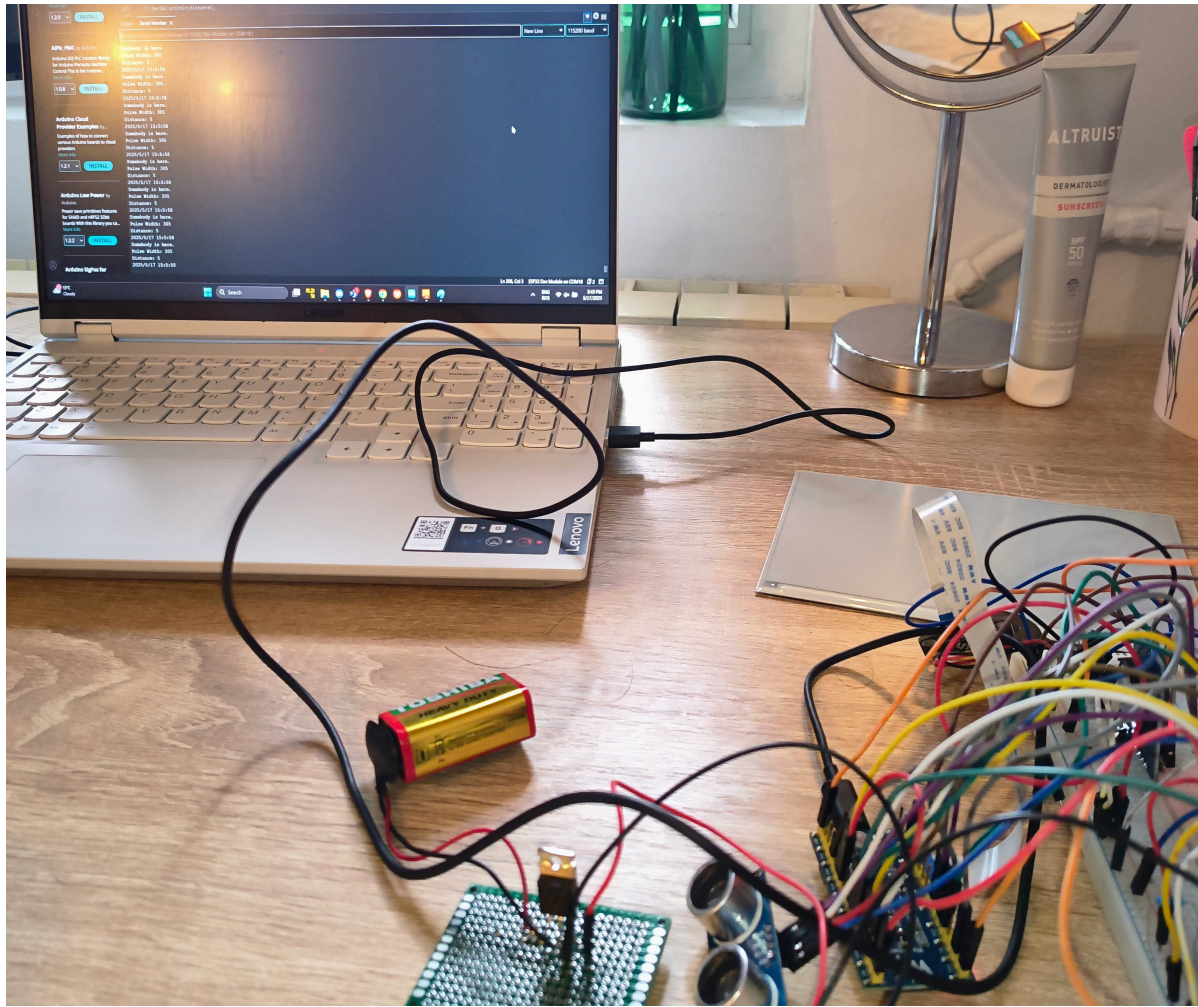
Am consultat pinout-ul plăcii mele ESP32 pentru a înțelege ce funcționalități oferă fiecare pin și am realizat conexiunile în funcție de protocoalele suportate de aceștia (ADC, PWM, I2C, SPI, UART). Astfel, fiecare componentă a fost legată la un pin corespunzător, asigurând compatibilitatea cu cerințele de comunicație sau control. Pentru difuzor am realizat un mic circuit de amplificare, descris în schema de mai jos, utilizând un tranzistor NPN, un condensator și câteva rezistențe, pentru a putea reda sunetul cu o intensitate adecvată. La butoane am adăugat rezistențe de tip pull-up pentru a evita fluctuațiile de semnal și pentru a asigura o detecție stabilă a apăsării. De asemenea, am inclus un comutator de tensiune de la 9V la 5V, astfel încât să pot alimenta componentele sensibile la supratensiune într-un mod sigur și eficient.



Circuitul înainte de mutare pe placa de prototipare:



Se observa in a 2 a poza ca functioneaza modulul RTC si ultrasonicul.



Software Design

Mediul de dezvoltare utilizat este Arduino IDE, împreună cu o placă ESP32 și un display Waveshare e-Paper de 7.5" (versiunea V2). Proiectul include următoarele biblioteci și fișiere:

- pentru RTC: RTCLib.h
- pentru senzorul ultrasonic și controlul pinilor: SPI.h, Wire.h
- pentru e-paper: DEV_Config.h, EPD.h, GUI_Paint.h, imagedata.h
- pentru buzzer: pitches.h, împreună cu un vector personalizat de frecvențe (Color.h) ce definește notele melodiei
- pentru citiri analogice și întreruperi: funcționalități native ESP32 (attachInterrupt, analogRead, pulseIn, etc.)
- pentru Bluetooth clasic (SPP): BluetoothSerial.h

Funcționalități principale Proiectul se bazează pe un sistem interactiv de detectare și notificare, folosind un e-paper display, un senzor ultrasonic, un fotorezistor, un buzzer și două butoane fizice. Interacțiunea dintre componente este realizată prin intermediul întreruperilor și a unor timere software.

1. Afișare oră și dată actualizată Display-ul e-paper afișează în mod implicit ora și data curente, preluate de la modulul RTC (DS3231). Informațiile se actualizează la fiecare minut nou, fără a rescrie constant ecranul, pentru a prelungi durata de viață a afișajului și pentru a preveni ghosting-ul.

2. Jurnal de evenimente (mișcare detectată) Cu ajutorul senzorului ultrasonic (HC-SR04), proiectul detectează obiecte aflate la mai puțin de 50 cm. Fiecare detecție este înregistrată cu timestamp complet (data și ora) în jurnalul de evenimente (până la 20 de intrări). Evenimentele sunt afișate pe e-paper la apăsarea butonului 1.

Pentru a preveni detecții excesive, o detecție este considerată validă doar dacă au trecut cel puțin 10 secunde de la ultima înregistrare (folosind `millis()`).

3. Afișare mesaj ambiental și melodie Dacă fotorezistorul detectează o luminozitate ridicată (valoare $ADC > 1000$), sistemul consideră că este „zi” și va:

- porni buzzer-ul cu o melodie predefinită
- afișa mesajul „Afară e zi!” pe e-paper
- alterna ecranul alb și mesajul la intervale de 5 secunde
- repeta melodia la infinit, cu pauze de 10 secunde între cicluri, până când se apasă butonul 2, caz în care melodia este oprită.

4. Control cu butoane fizice

- Buton 1: alternează între afișarea jurnalului de mișcări și revenirea la ora curentă. La revenire, jurnalul este resetat (șters complet).
- Buton 2: în orice moment, dacă este apăsat în timpul redării unei melodii, oprește imediat melodia și revine controlul la funcționarea normală.

5. Trimitere jurnal prin Bluetooth către aplicația mobilă Aceasta permite trimiterea evenimentelor înregistrate către o aplicație Android.

- ESP32 se identifică prin Bluetooth ca ESP32_BT
- Aplicația mobilă se conectează la el și poate primi date
- La apăsarea butonului 2, ESP32 trimite toate logurile de mișcare stocate (maximum 20), fiecare sub formă de text formatat, de exemplu: Mișcare: 24/05/2025 17:03:12

Funcția `setup()` În această funcție se realizează:

- inițializarea pinilor de intrare/ieșire pentru senzori și butoane
- atașarea întreruperilor pentru cele două butoane și pentru pinul SQW de la RTC
- inițializarea RTC și setarea alarmei pe un orar fix
- inițializarea e-paper și alocarea memoriei RAM pentru imagine
- afișarea datei și orei curente

Funcția `loop()` Fluxul principal este:

- Se citește valoarea fotorezistorului. Dacă e lumină puternică, se pornește rutina de melodie și afișare.
- Se verifică dacă s-a apăsător butonul 1 sau butonul 2 și se execută comportamentul asociat.
- Se citește distanța de la senzorul ultrasonic și, dacă este validă, se loghează un nou eveniment și se pornește melodia.
- Se verifică dacă este un minut nou și, dacă da, se actualizează ora afișată.
- Se fac printări în serial pentru debug (cu data/ora și distanță).

Interfața serială Pentru debugging s-au folosit mesaje `Serial.println()` pentru urmărirea:

- detecțiilor cu timestamp
- comutărilor între stări
- declanșării sau opririi melodiei

Rezultate Obținute

Videoclip proiect

<https://youtube.com/shorts/j-p8dAod7mA?feature=share>

Concluzii

Download

<https://github.com/CezarVulpe/Proiect-PM>

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2025/aluca/cezar_andrei.vulpe



Last update: **2025/05/26 02:30**