

Security System

Autor: [Laurențiu-Constantin Petre](#)

Grupa: 333CA

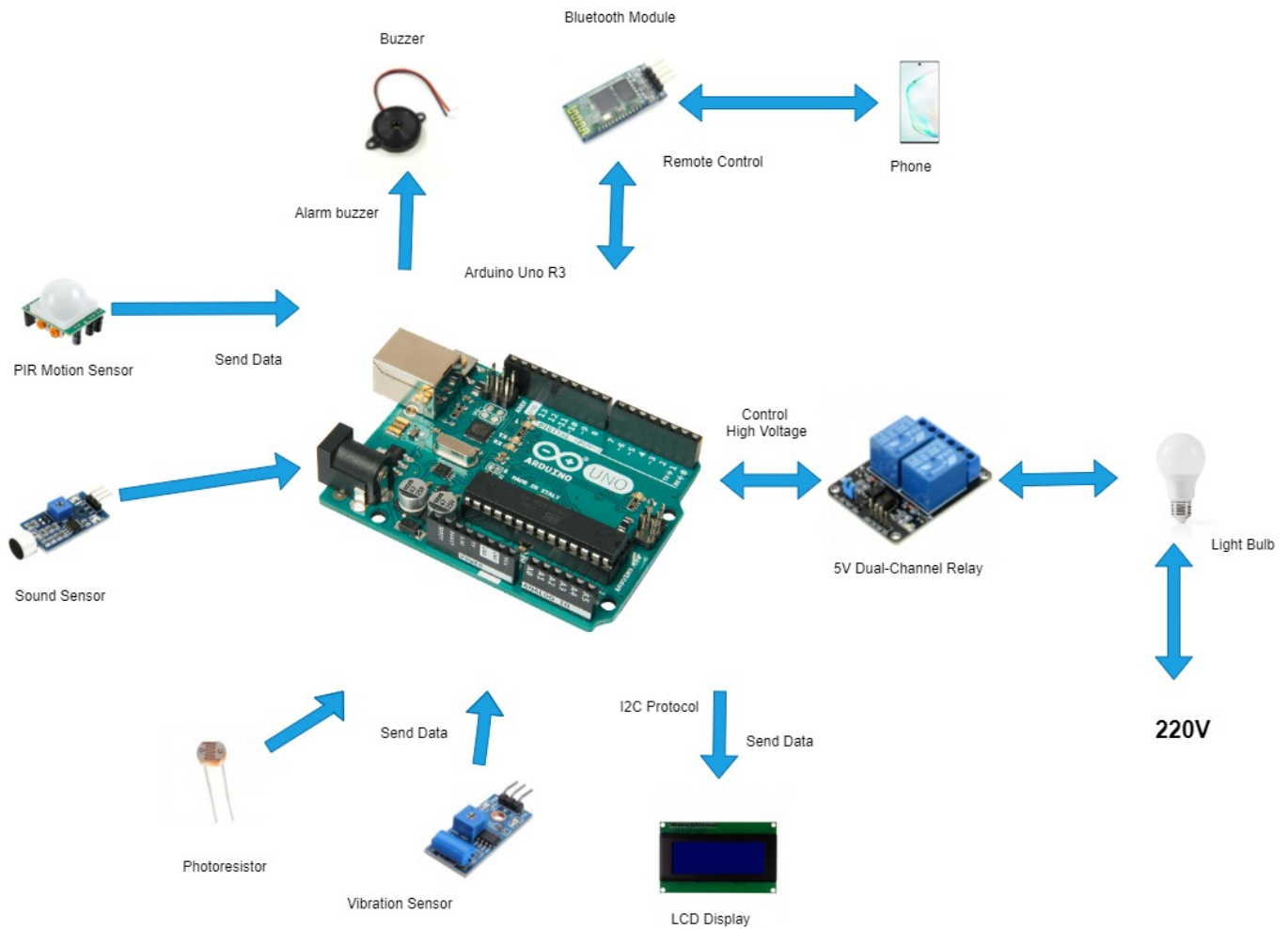
Introducere

- În cadrul proiectului, voi implementa un sistem de avertizare bazat pe o rețea formată din dispozitive tehnice (senzori).
- **Scopul acestui proiect** este de a putea intercepta diferite evenimente care au loc, (o explozie, un cutremur, o spargere a unei case) și de a putea notifica proprietarii de bunuri economice, în acest caz.
- **Ideea principală** a sistemului este de protecție a diferitelor obiecte, care nu pot fi supravegheate în mod continuu de proprietari, fapt ce a condus la aceste tehnologii bazate pe senzori.
- **Utilitatea principală** derivă din necesitatea apariției acestor sisteme de protecție, fiind un bun pion în detecția evenimentelor neplăcute.

Descriere Generală

- Prin intermediul modului bluetooth se pot trimite comenzi de aprindere sau stingere a becului de la un dispozitiv Android.
- Se detectează obiectele din apropiere cu ajutorul senzorului de mișcare și în funcție de intensitatea luminii se poate semnaliza prin aprinderea becului într-un mediu nocturn.
- Se folosește un ecran LCD pentru a se afișa dacă a detectat vreun obiect sau ce comandă a fost trimisă prin bluetooth.
- Pentru a putea schimba starea becului, folosesc un releu care deschide sau închide circuitul format din bec și tensiunea de alimentare de 220V.

Schema bloc



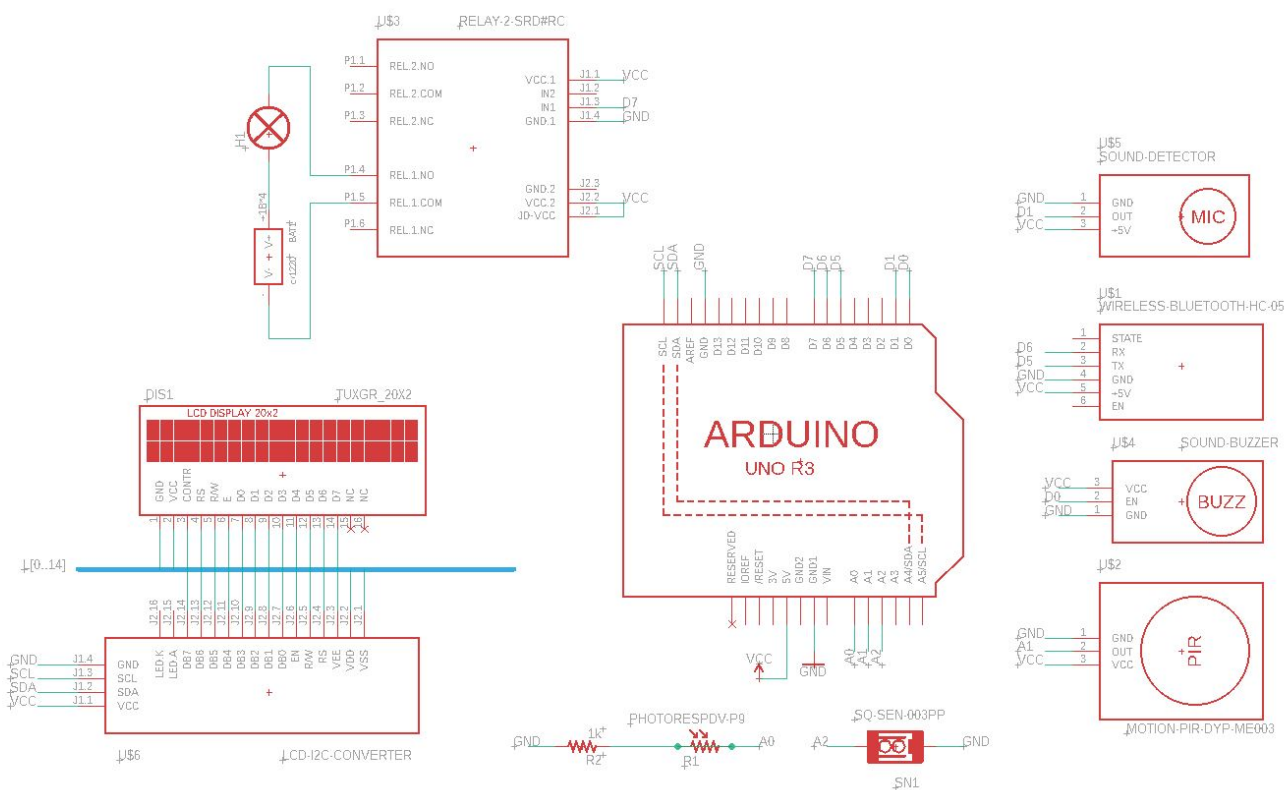
Hardware Design

Componente folosite

- Arduino UNO R3
- Breadboard
- Leduri
- Becuri
- Senzor de lumină
- Senzor de sunet
- Senzor de vibrație
- Rezistențe
- Releu
- Fire
- Ecran LCD
- Fotorezistor
- Buzzer

- Modul Bluetooth

Schema electrică



Software Design

Software utilizat în cadrul proiectului

- [Arduino IDE](#) → Mediul de programare.
- [LiquidCrystal-I2C](#) → Biblioteca folosită pentru comunicarea cu LCD.
- [SoftwareSerial.h](#) → Biblioteca folosită pentru comunicarea prin bluetooth.
- [EAGLE](#) → Software pentru realizarea schemei electrice.
- [APP.DIAGRAMS](#) → Software pentru realizarea schemei bloc.

Despre variabilele folosite

- **BTSerial**, o instanță pentru comunicarea prin bluetooth cu smartphone - ul.
- **lcd**, o instanță pentru comunicarea prin I2C cu LCD - ul.
- **t_vibr**, **t_sound**, **t_light**, timpii folosiți pentru prelucrarea output - urilor.
- **flagVibration**, **flagSound**, **flagLight**, flaguri folosite pentru a afișa mesaje corespunzătoare în cazul detecțiilor de factori externi.
- **blinkBulb**, **timeBlinkBulb**, variabile folosite pentru a putea aprinde/stinge becul (alternează).
- **valueLight**, **valuePIR**, variabile folosite pentru a stoca valoarea dată de senzorul de lumină, respectiv mișcare.
- **stateBuzzer**, variabilă folosită pentru a putea realiza semnale sonore în urma pornirii buzzer - ului printr-o comandă.

Sunt implementate câteva comenzi pe care le pot recepționa de la smartphone:

- BLINK BULB X → comută becul din stare aprins și stins la X secunde
- TURN ON BULB → stinge becul
- TURN OFF BULB → aprinde becul
- BUZZER ON → activează modulul buzzer, bazat pe senzorul de mișcare
- BUZZER OFF → dezactivează modulul buzzer
- ENABLE PIR → activează senzorul de mișcare
- DISABLE PIR → dezactivează senzorul de mișcare

Despre funcțiile implementate

Funcția **setup()**:

- Se initializează modul de lucru al pinilor folosiți INPUT/OUTPUT.
- Se trece releul în starea OPENED, deoarece se folosește modul Normally Open (NO).
- Se inițializează comunicarea cu interfața serială și cea a modulului bluetooth.
- Se inițializează interfața LCD - ului.

Funcția **loop()**:

- Se apelează funcțiile pentru detecția factorilor (de sunet, lumină, mișcare, vibrație).
- Se apelează funcția ce se ocupă cu recepția comenzilor și comunicarea prin bluetooth.

Funcția **detectMotion()**:

- Se verifică dacă sunt activate prelucrările din cadrul funcției.
- Se citește intrarea de pe pinul corespunzător senzorului de mișcare.
- Se aprinde becul și pornește buzzer - ul (cu starea on) dacă se detectează un obiect.

Funcția **detectVibration()**:

- Se citește intrarea de pe pinul corespunzător senzorului de vibrație.
- Se calculează timpul în milisecunde pentru a se ști când se poate dezactiva output - ul de pe LCD.

- Dacă nu se vibrează, se golește linia corespunzătoare de pe LCD, altfel se afișează un mesaj specific.

Funcția **detectSound()**:

- Se citește intrarea de pe pinul corespunzător senzorului de sunet.
- Se calculează timpul în milisecunde pentru a se știi când se poate dezactiva output - ul de pe LCD.
- Se verifică dacă vreun obiect scoate semnale sonore, afișându-se un mesaj corespunzător în caz de detecției cu succes.

Funcția **detectLight()**:

- Se citește intrarea de pe pinul corespunzător senzorului de lumină.
- Se calculează timpul în milisecunde pentru a se știi când se poate dezactiva output - ul de pe LCD.
- Dacă se detectează lumină, se afișează un mesaj corespunzător.

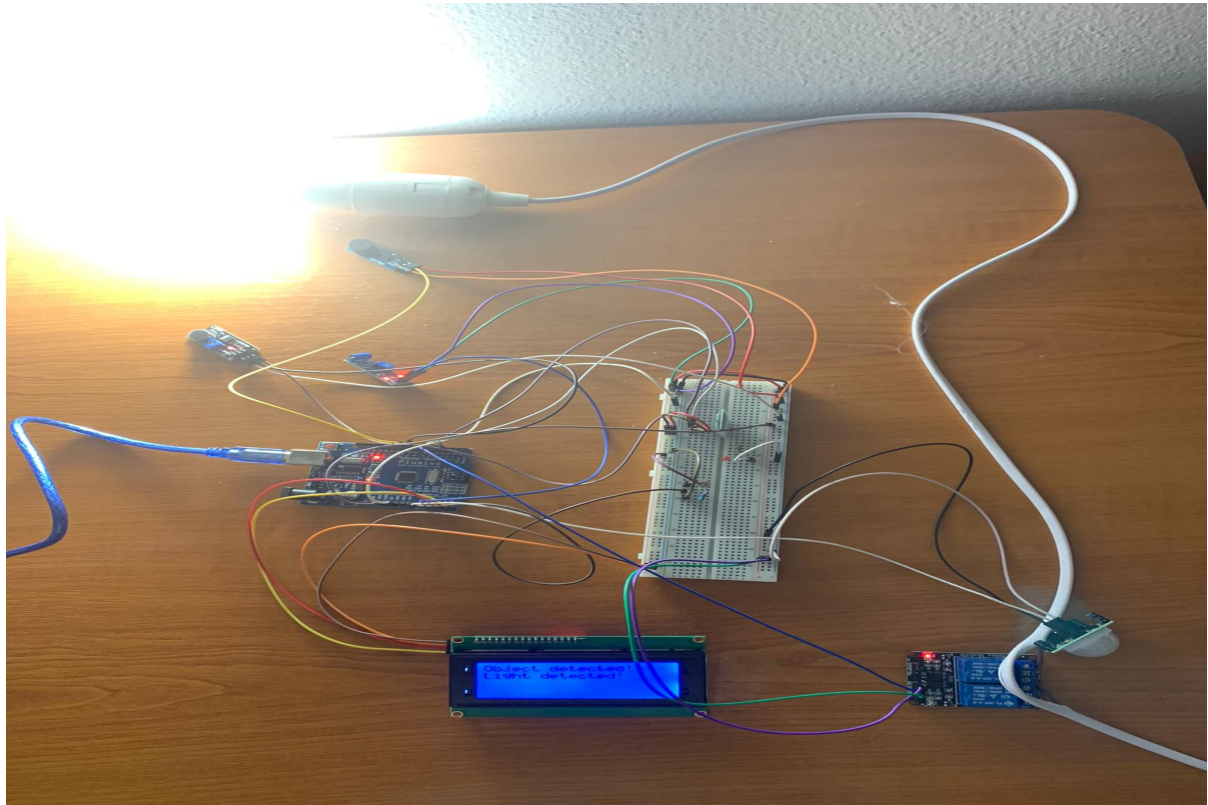
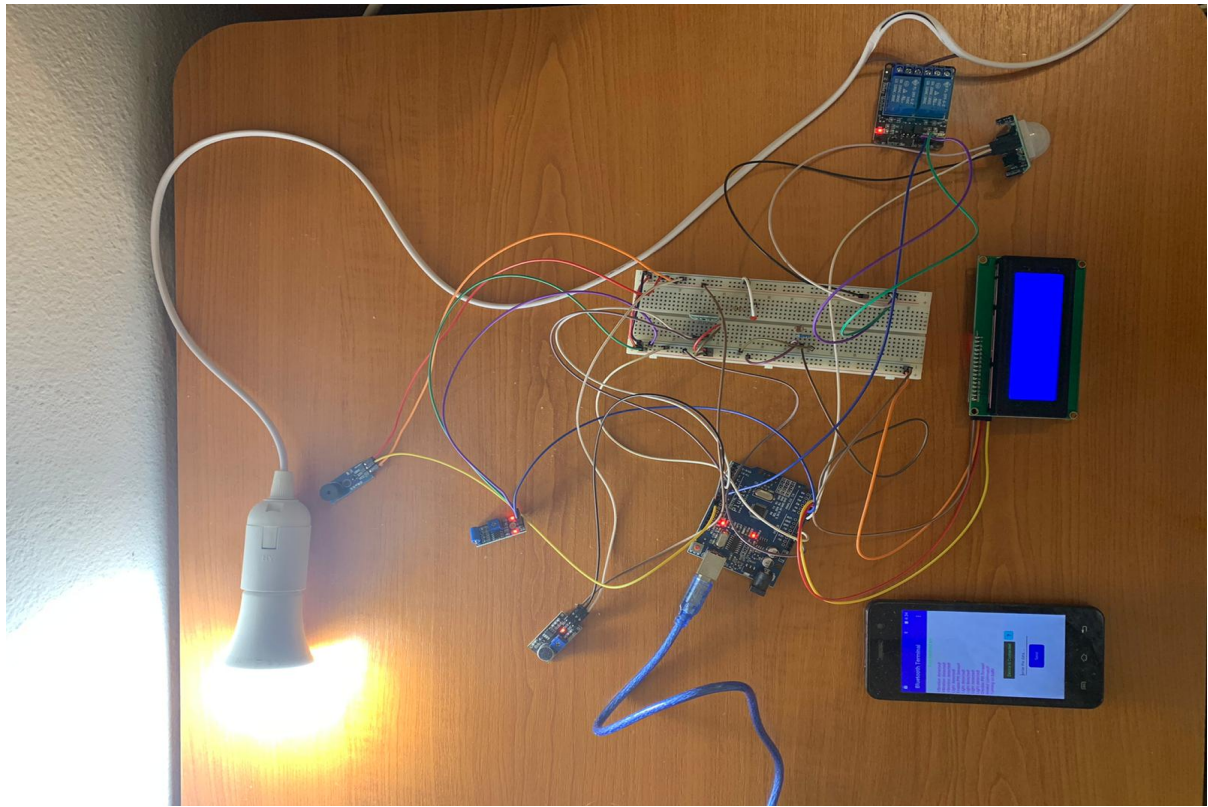
Mesajele către LCD, sunt transmise și către smartphone prin bluetooth.

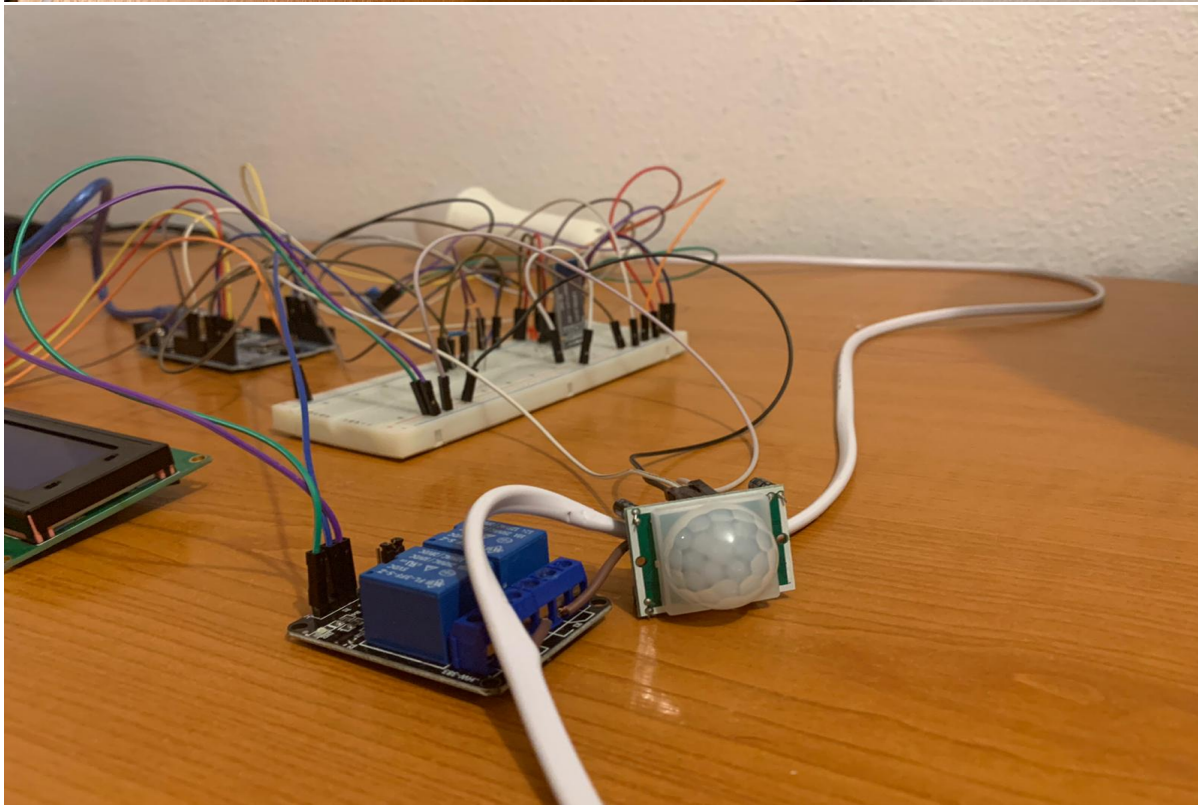
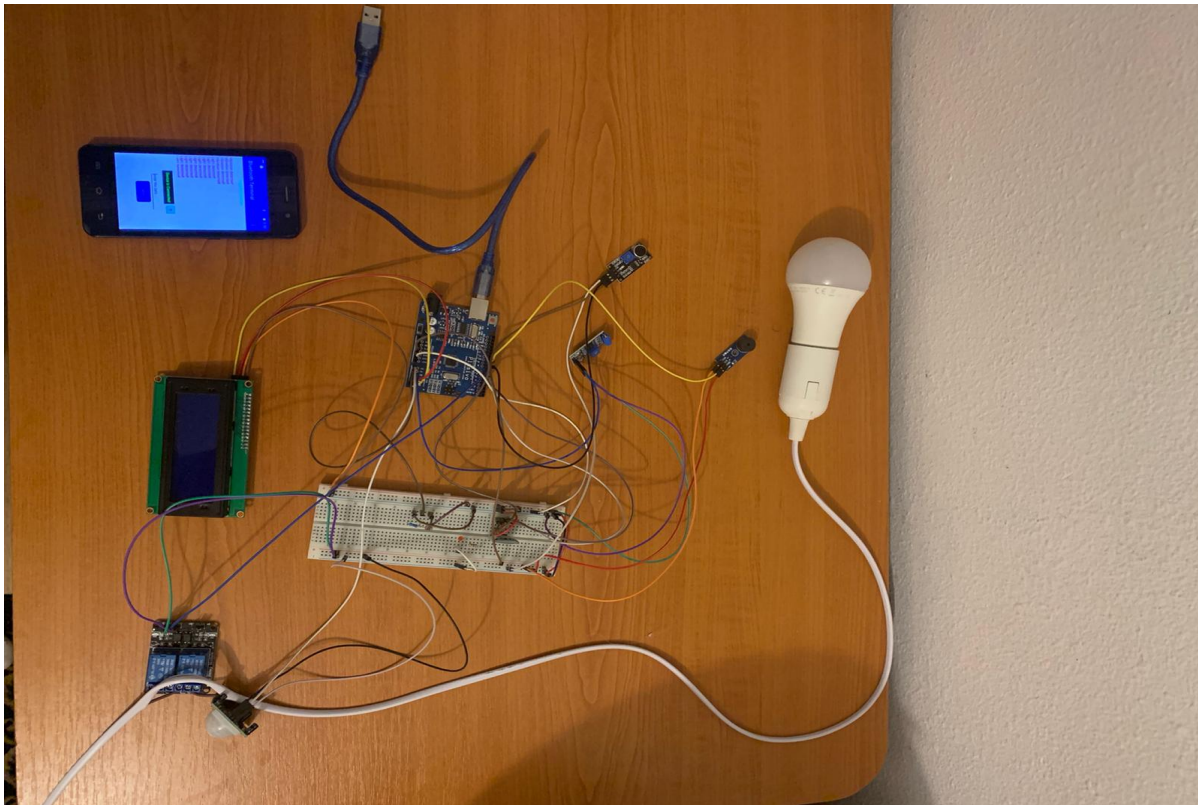
Funcția **communicationViaBT()**:

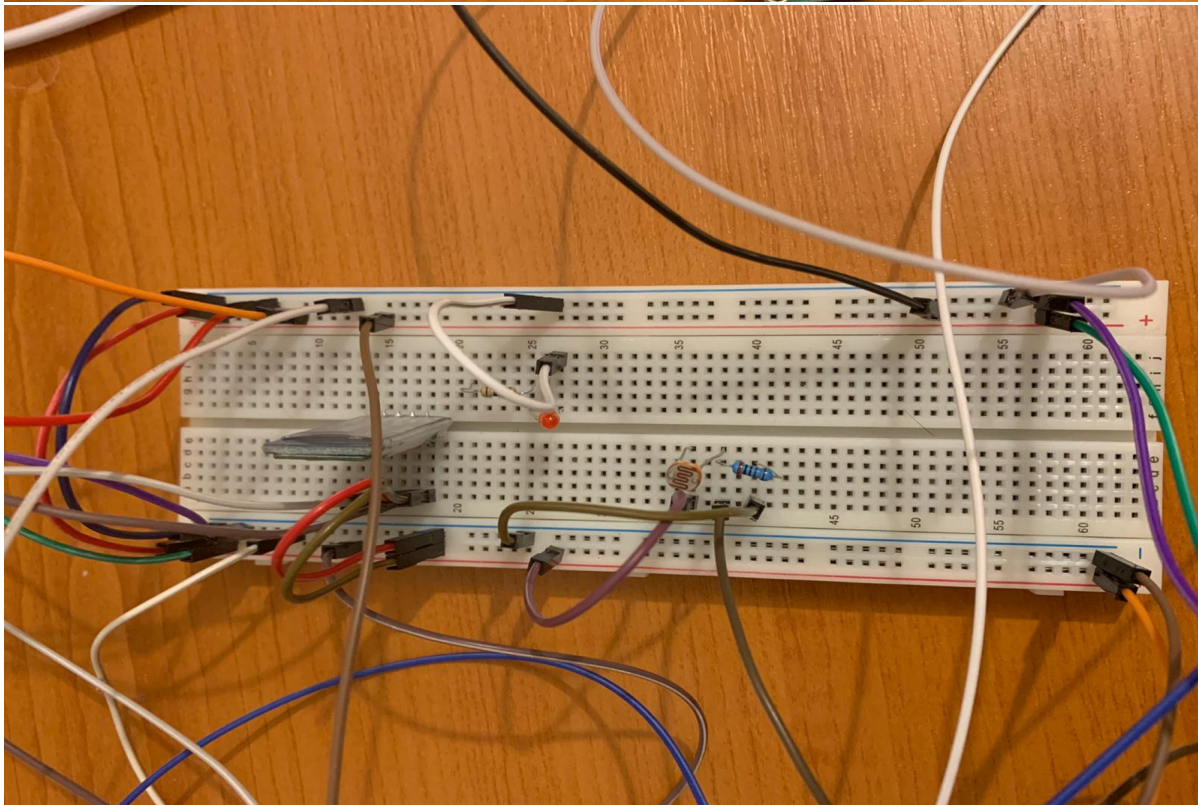
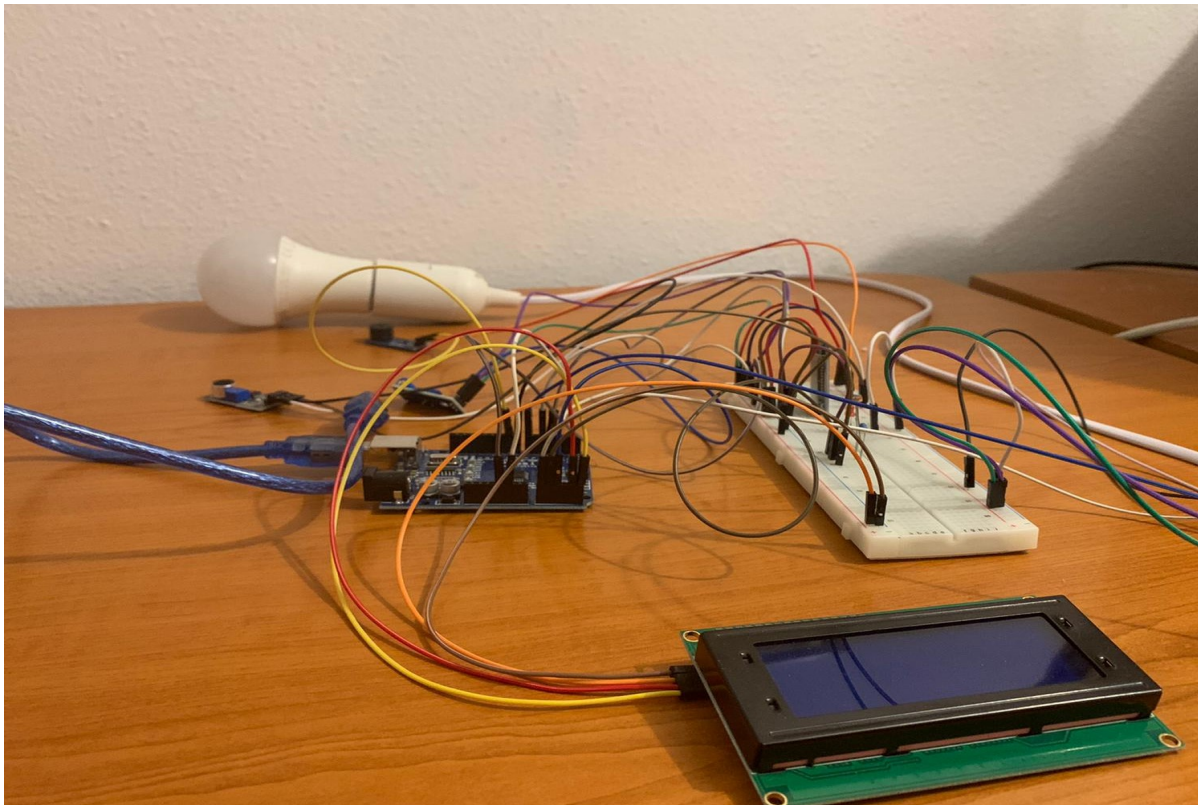
- Se verifică dacă este disponibil o comandă dată prin bluetooth.
- Se verifică dacă este o comandă validă și se afișează un mesaj corespunzător în cazul în care este invalidă.
- Se compară cu fiecare caz de comandă validă și se prelucrează corespunzător comanda și mesajul de output.

Rezultate Obținute

Imagini cu componentele proiectului







Videoclipul demonstrativ

- [Link Demo](#)

Concluzii

- Prin acest proiect, am înțeles cum se folosește o plăcuță cu microcontroller conectată la diferiți senzori.
- Am realizat că nu există, cel mai probabil, aplicații mobile care să folosească tehnologia **BLE** pentru comunicarea prin bluetooth cu smartphone - uri ce folosesc IOS.
- Am fost nevoit să folosesc pentru senzorul de mișcare, doar modul de funcționare Single Trigger Mode.
- Pentru alimentarea becului, am folosit tensiunea de 220V deoarece nu am găsit un bec cu putere mare pentru tensiunile mici de la baterii.
- Am remarcat că fotorezistorul se degradează foarte repede dacă este ținut la lumină zilnic, fapt pentru care am cumpărat rezerve.
- Există posibilitatea să te curentezi sau mai grav de atât, dacă nu știi să folosești releul, deoarece sunt 220V, iar omul nu rezistă la această tensiune.
- Piese din China, câteodată, își fac simțită calitatea deoarece am avut probleme cu recunoașterea plăcuței Arduino, dar și cu câțiva senzori care se încingeau destul de puternic.
- Potențiometrele de la componente se pot strica repede. Deci, trebuie să avem mare atenție când lucrăm cu ele.
- Plăcuța Arduino Uno are un dezavantaj: are prea puțini pini analogici, 6, în comparație cu alte plăcuțe (Arduino Mega → 16 pini analogici).
- Am învățat multe lucruri utile care îmi vor fi de folos în viitor. Acest proiect se poate extinde foarte mult, utilizând senzori mai eficienți și precizi în detecția de evenimente nedorite. După cum se observă, este folosit în industrie foarte mult. Oricine își dorește o bună monitorizare a bunurilor mobile și imobile, apelează la un sistem bazat pe supraveghere video, audio, dar și pe detecția mișcării sau a vibrațiilor (în cazul cutremurelor).

Download

- Link Github: [Security System](#)
- Arhiva ce conține codul sursă: [333ca_petre_laurentiu-constantin_proiect_pm.zip](#)

Jurnal

- 28 Aprilie 2021 → Alegerea proiectului
- 30 Aprilie 2021 → Achiziție componente principale(senzori, module)
- 5 Mai 2021 → Achiziție LCD, modul bluetooth.
- 9 Mai 2021 → Achiziție modul pentru bec
- 1 Iunie 2021 → Creare video demo despre proiect
- 2 Iunie 2021 → Concluzii
- 2 Iunie 2021 → Finalizare pagina de wiki

Bibliografie/Resurse

PDF: [Security-System](#)

Pagină Wiki: [Link](#)

Tutoriale: [Releu](#) , [Senzor de mișcare](#) , [Fotorezistor](#) , [Modul Bluetooth](#)

Aplicație folosită: [Bluetooth Terminal: for your Arduino](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/cghenea/security-system>



Last update: **2021/06/04 21:41**