

JewelGuard Proximity Defender

Introducere

Proiectul reprezintă un sistem inovator de securitate pentru protejarea obiectelor de valoare, precum bijuteriile. Acesta utilizează tehnologia avansată de detectare a mișcării pentru a monitoriza în timp real orice mișcare suspectă în apropierea obiectului protejat.

Scopul acestui proiect este de a preveni furtul și de a oferi utilizatorilor un sentiment sporit de siguranță în ceea ce privește protejarea obiectelor lor de valoare.

Am pornit de la ideea că în cămin la o petrecere organizată la mine în camera cineva a furat una din bijuteriile mele. Soluțiile tradiționale de securitate nu sunt tocmai accesibile, așa că un astfel de mecanism simplu ar fi fost util pentru a mă avertiza și a afla cine a comis fapta.

Proiectul este util și pentru alții deoarece poate fi adaptat pentru a proteja o varietate de obiecte valoroase sau importante, nu numai bijuterii. Acesta combină tehnologia de detectare a mișcării cu funcții avansate precum alertele în cascada, mecanismul de panică și camera de supraveghere pentru a asigura o protecție robustă și eficientă.

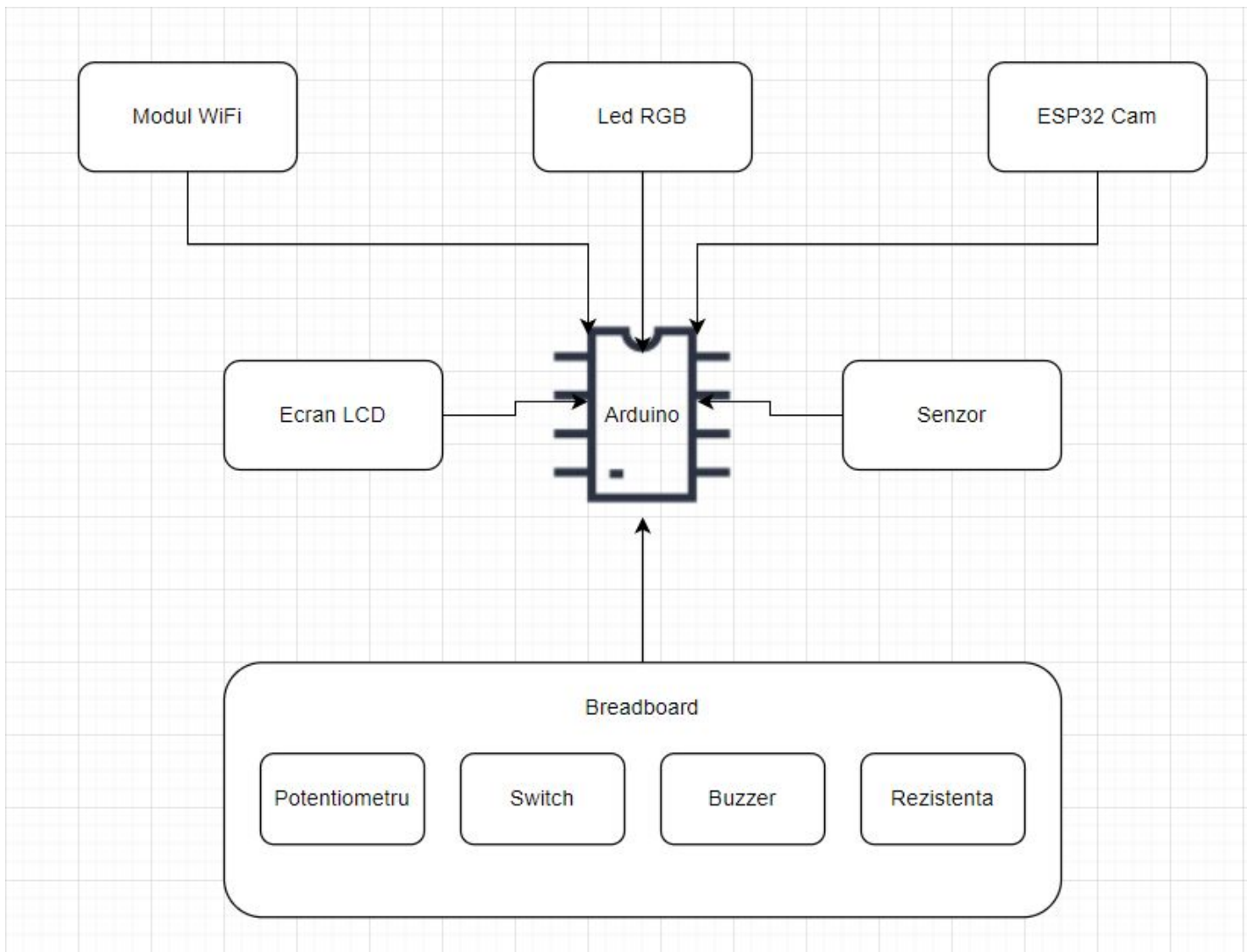
Descriere generală

1. Arduino Uno: Este o placă de dezvoltare bazată pe microcontrolerul ATmega328P. "Creierul" proiectului pentru a controla și coordona funcționarea tuturor celorlalte componente.
2. Breadboard: O plăcuță de conexiuni, fără a necesita lipirea componentelor.
3. Ecran LCD: Un ecran pentru afișarea stării sistemului și a informațiilor relevante pentru utilizator.
4. Buzzer: Un dispozitiv care emite sunete pentru a alerta utilizatorul în cazul unei încălcări a securității.
5. Potentiometru: O componentă electronică reglabilă care permite ajustarea nivelului de sensibilitate al sensorului de distanță și al ecranului LCD.
6. Rezistori: Componente pasive care limitează sau reglează curentul în circuit.
7. Switch: Un întrerupător folosit pentru a activa sau dezactiva sistemul.
8. Led RGB: Un LED care poate emite diferite culori, utilizat pentru a indica starea sistemului prin intermediul culorilor diferite.
9. Sensor de distanță: Un dispozitiv care măsoară distanța până la obiectele din apropiere, utilizat pentru a detecta mișcarea în jurul obiectului protejat.
10. Placa WiFi: Permite comunicarea wireless între Arduino și ESP32 Cam pentru a transmite datele și a controla camera.
11. ESP32 Cam: O cameră WiFi compactă și de înaltă rezoluție care capturează și trimite fotografii în cazul unei încălcări a securității.
12. Modul GSM: Un modul de comunicare celulară care permite trimiterea de mesaje text și apeluri în cazul unei alerte de securitate.

În cadrul sistemului, Arduino Uno controlează majoritatea componentelor, inclusiv ecranul LCD, buzzerul, sensorul de distanță. Sensorul de distanță monitorizează în mod constant mediul înconjurător, iar în cazul detectării unei mișcări suspecte, buzzerul și LED-ul RGB sunt activate pentru a emite alerte. În același timp, ESP32 Cam face o fotografie a potențialului intrus și o trimite împreună cu un mesaj prin intermediul aplicației Telegram către telefonul mobil al proprietarului. Acesta se face utilizând API-ul oferit de aplicație pentru a permite comunicația cu telefonul utilizatorului.

La activare, ecranul LCD va afișa un mesaj care indică faptul că sistemul a fost pornit și sensorul va începe să detecteze mișcările din împrejurimi. Nivelul de avertizare detectat de sensor va fi afișat atât pe ecranul LCD, cât și pe LED-ul RGB. În cazul în care suspectul este prea aproape de obiect și probabil va încerca să îl fure, sistemul va trimite o poză și un mesaj proprietarului.

Schema bloc

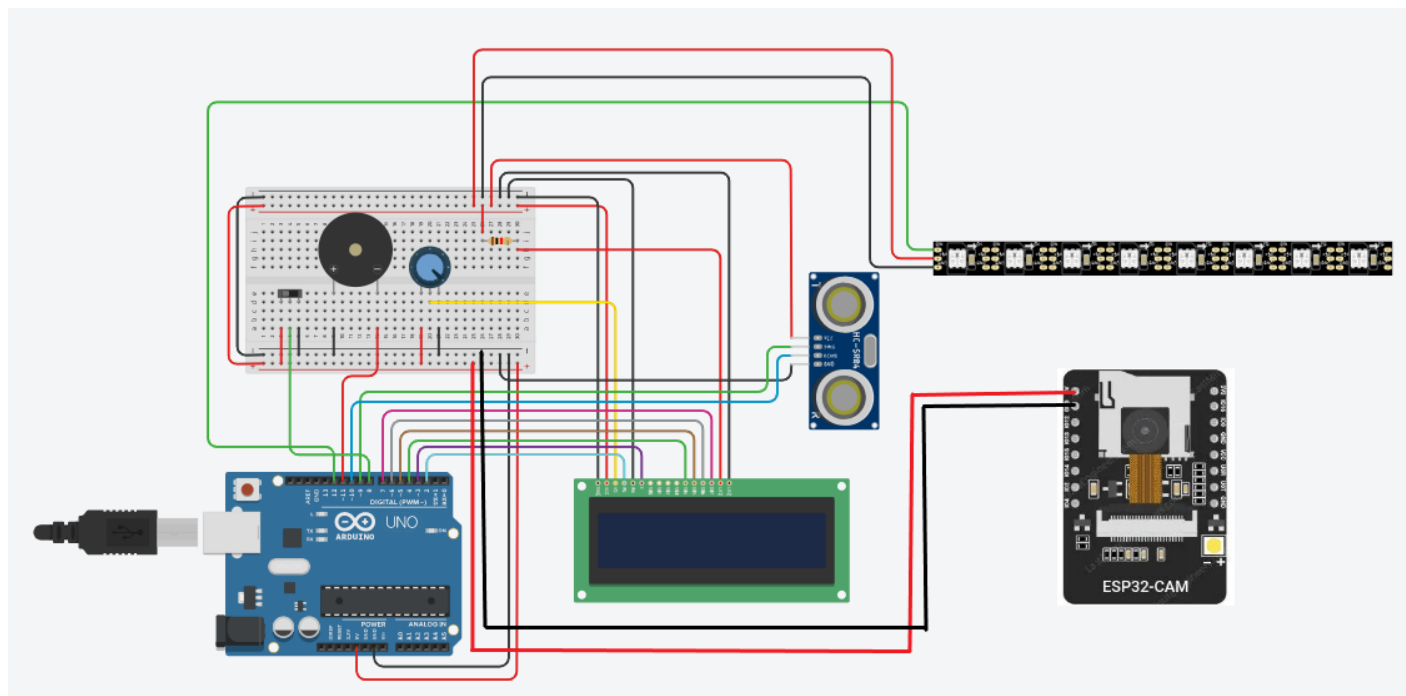


Hardware Design

Compenente

- Arduino Uno
- Breadboard
- Ecran LCD
- Buzzer
- Potentiometru
- Rezistori
- Switch
- Led RGB
- Senzor distanta
- Placa WiFi
- ESP32 Cam

Schema electrica



Software Design

Pentru dezvoltarea proiectului am folosit Arduino IDE si librariile:

- LiquidCrystal.h - display-ului LCD
- Adafruit_NeoPixel.h - banda LED RGB
- WiFi.h - conectarea placutei ESP32Cam la Wifi-ul personal
- UniversalTelegramBot.h - folosirea API-ului oferit de Telegram
- ArduinoJson.h - <https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot>
[<https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot>]

În cadrul funcției loop(), codul verifică mai întâi starea comutatorului conectat la switchPin. Dacă acesta se află în starea LOW, înseamnă că sistemul este oprit. În acest caz, ecranul LCD afișează un mesaj care indică faptul că sistemul este oprit.

Dacă comutatorul se află în starea HIGH, ceea ce indică faptul că sistemul este pornit, codul continuă cu logica de măsurare a distanței și de afișare.

În primul rând, banda de LED-uri este pornită secvențial pentru fiecare LED, creând un efect de defilare.

```
int sine[] = {0, 1, 2, 3, 4, 5, 6, 7};
for(int i = 0; i < 8; i++) {
  strip.setPixelColor(sine[i], 0);
  strip.setPixelColor(sine[(i + 7) % 8], color);
  strip.show();
  delay(150);
}
```

Aceasta stabilește culoarea a doi pixeli adiacenți la un moment dat. Funcția setPixelColor() este apelată cu sine[i] pentru a dezactiva un pixel și cu sine[(i + 7) % 8] pentru a seta culoarea pe următorul pixel din secvență. Acest lucru creează efectul de defilare a culorii prin banda de LED-uri. Funcția strip.show() este apelată pentru a actualiza banda de LED-uri cu noua configurație de culoare, iar apoi există o întârziere de 150 de milisecunde înainte de a trece la următoarea iterație.

Apoi, senzorul cu ultrasunete este declanșat prin setarea trigPin-ului la LOW, întârzierea pentru o perioadă scurtă de timp și apoi setarea acestuia la HIGH timp de 10 microsecunde înainte de a-l seta din nou la LOW.

Durata impulsului de ecou se măsoară cu ajutorul funcției pulseIn() și se stochează în variabila duration. Distanța se calculează folosind durata și viteza sunetului în aer și se stochează în variabila distance.

Ecranul LCD este apoi actualizat cu măsurarea distanței prin imprimarea valorii distanței și a unității de măsură.

În continuare, codul verifică valoarea distanței pentru a determina culoarea corespunzătoare pentru banda LED și mesajul corespunzător care trebuie afișat pe LCD. Dacă distanța este mai mică de 50 cm, culoarea LED-ului devine roșie, iar pe LCD se afișează "DANGER". Buzzer-ul este pornit mai galagios pentru a oferi o alertă. Dacă distanța este cuprinsă între 50 cm și 100 cm, culoarea LED devine galbenă, iar pe ecranul LCD se afișează "SOMEONE". De asemenea, se activează ceva mai încet buzzer-ul. Dacă distanța este mai mare de 100 cm, culoarea LED-ului devine verde, iar pe ecranul LCD se afișează "NOBODY".

Bucula se repetă apoi, măsurând continuu distanța și actualizând afișajul pe baza valorii măsurate până când sistemul este oprit prin setarea comutatorului în starea LOW.

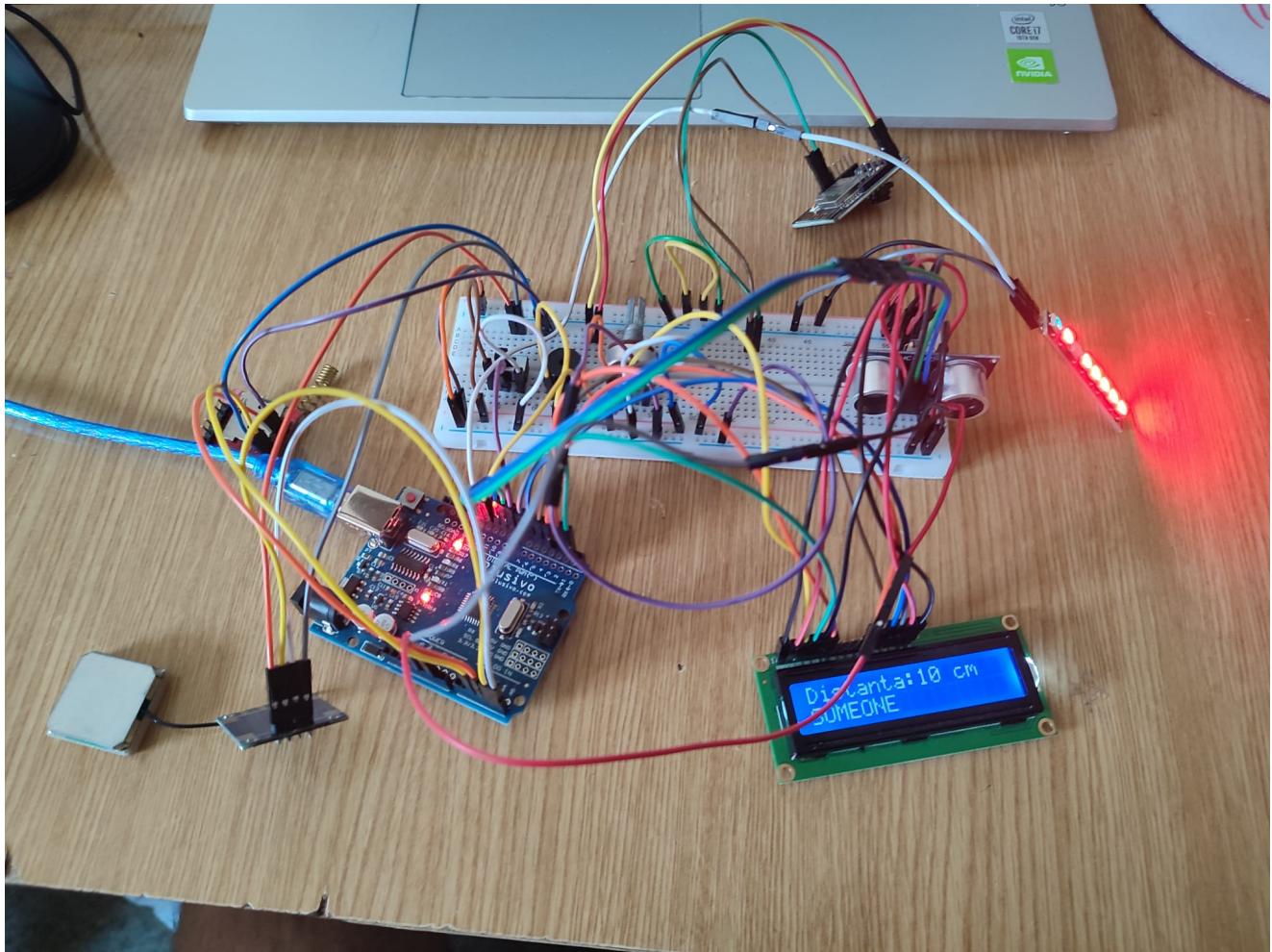
ESP32 CAM + Telegram

Pentru partea de configurare a placutei ESP32Cam am urmat pașii următori:

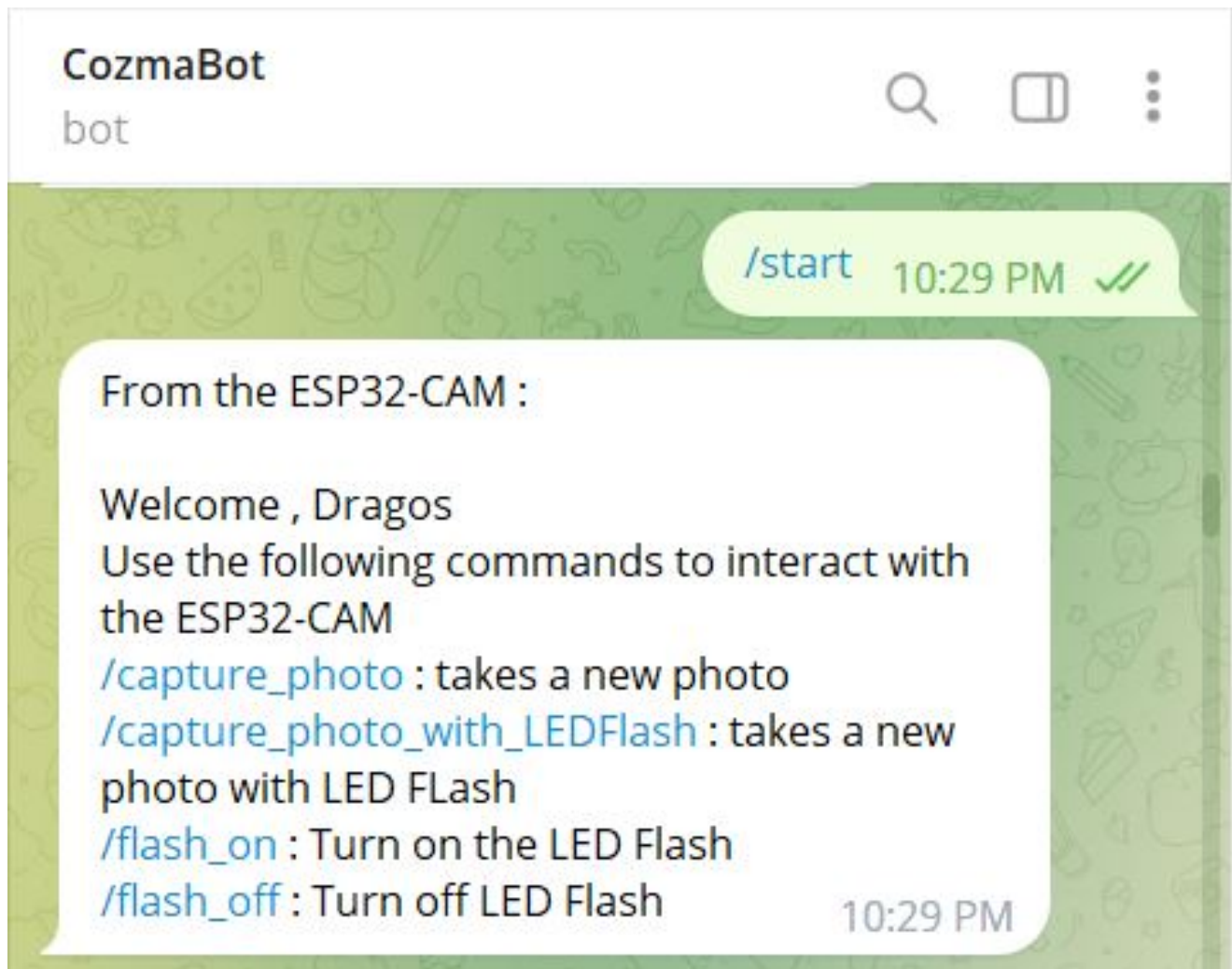
1. Crearea botului Telegram: Înainte de a utiliza codul, trebuie să creați un bot Telegram folosind botul BotFather pe Telegram. BotFather vă va furniza un token unic pentru botul dumneavoastră, pe care va trebui să îl includeți în cod ca variabilă BOTtoken.
2. Inițializarea rețelei: Codul stabilește o conexiune la rețeaua Wi-Fi folosind SSID-ul și parola furnizate. Odată conectat, acesta inițializează obiectul WiFiClientSecure, care este utilizat pentru a comunica în siguranță cu serverul Telegram API.
3. Recuperarea și manipularea mesajelor: Codul include o subrutină numită handleNewMessages, care este responsabilă pentru verificarea mesajelor noi primite de către Telegram bot. Aceasta interoghează periodic serverul Telegram API pentru a prelua orice mesaj nou. Dacă sunt găsite mesaje noi, codul le procesează în mod corespunzător.
4. Procesarea mesajelor: Atunci când se primește un mesaj nou, codul verifică conținutul mesajului. În acest cod, acesta verifică în mod specific dacă mesajul este "/photo" (indicând o cerere de trimitere a unei fotografii). Dacă mesajul se potrivește cu această comandă, codul apelează subrutina sendPhotoTelegram pentru a captura o fotografie cu ajutorul ESP32-CAM și a o trimite pe chat-ul specificat.
5. Trimiterea fotografiei: Cererea HTTP construită este trimisă către serverul Telegram API folosind metoda POST. Serverul procesează cererea și livrează fotografia către chat-ul specificat. Dacă fotografia este trimisă cu succes, se afișează un mesaj de succes. Dacă apare o eroare în timpul procesului de trimitere, se afișează un mesaj de eroare.

Rezultate Obținute

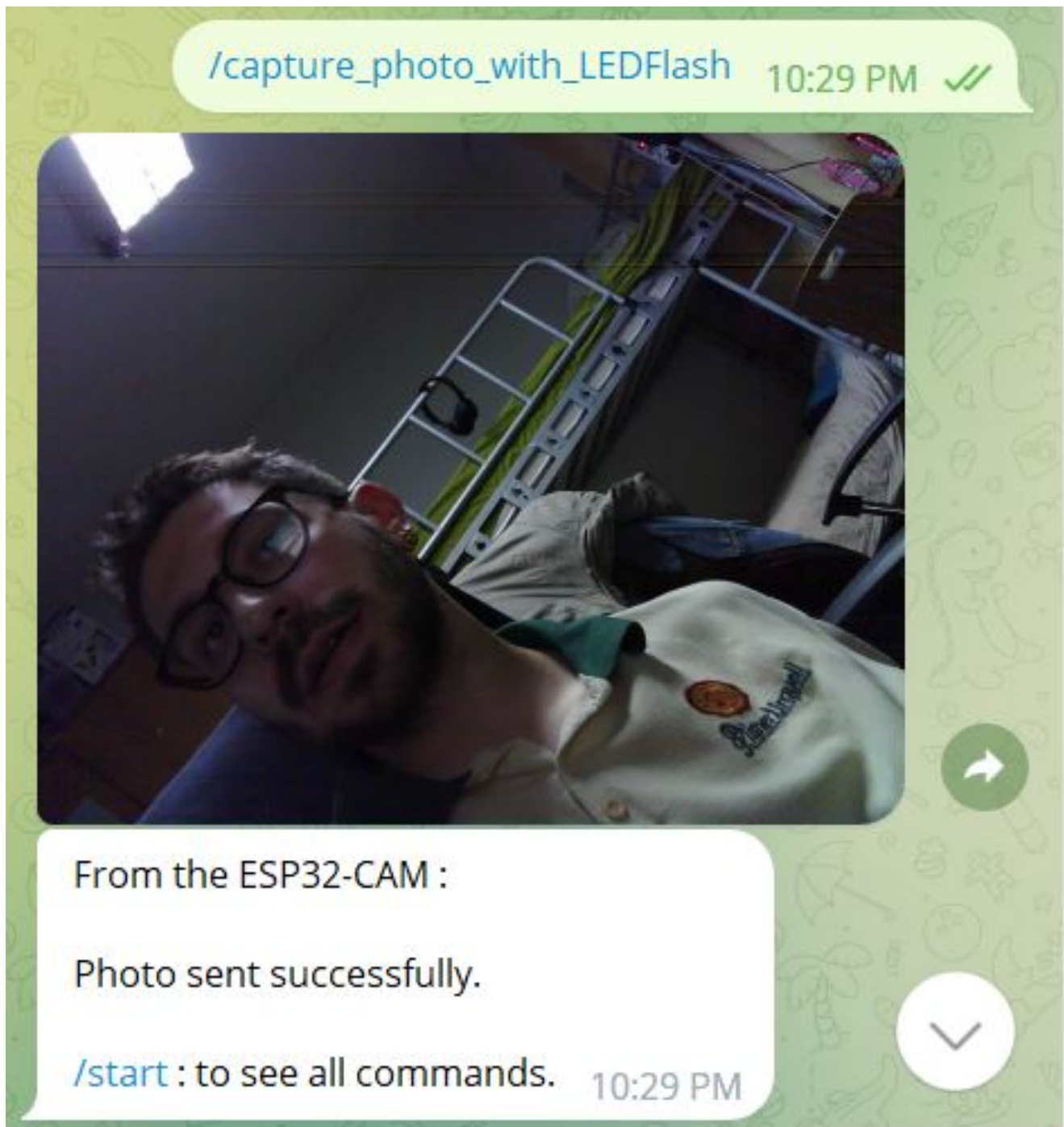
Prima iteratie a proiectului



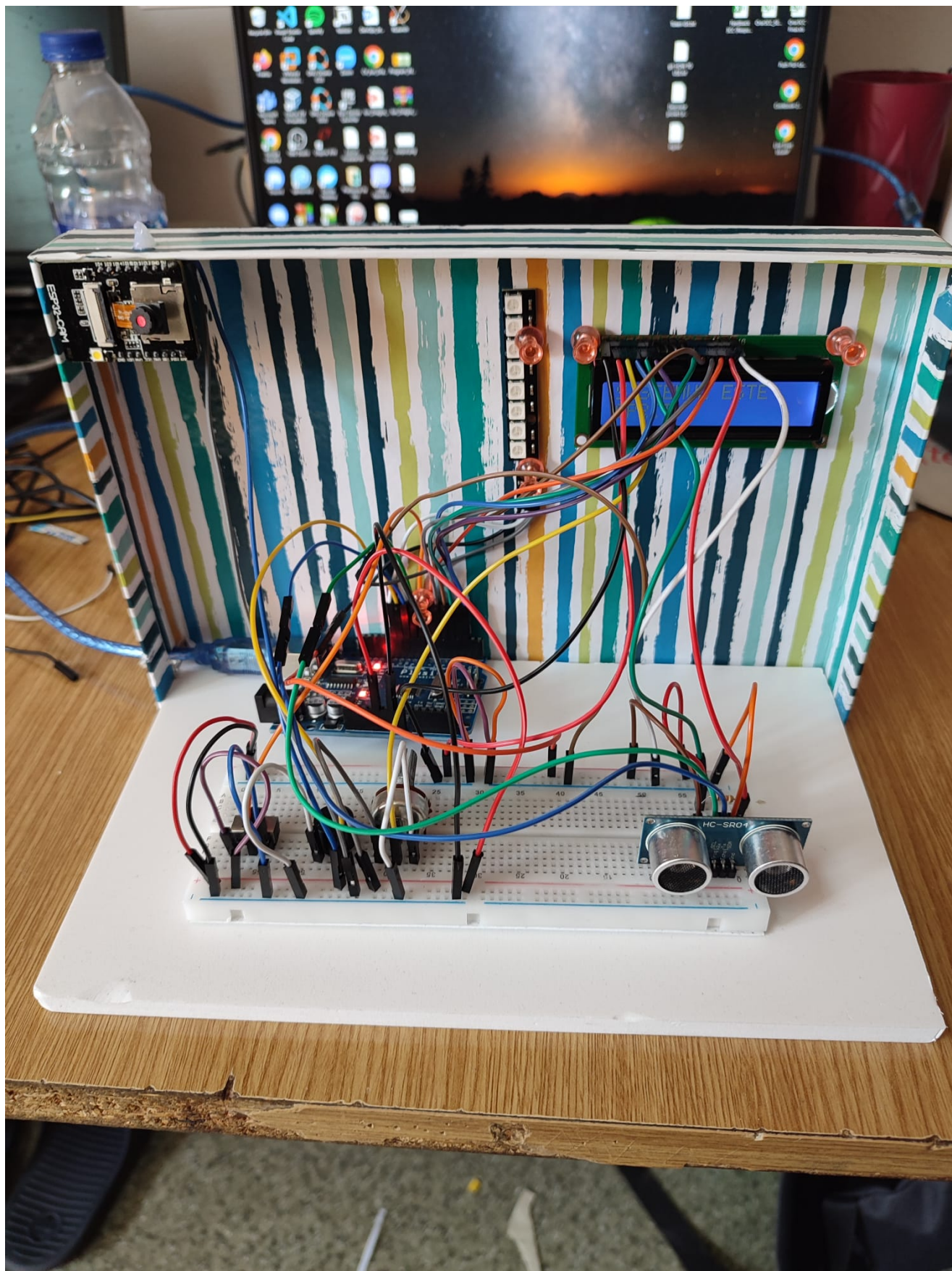
Comenzi Chat Telegram



Poze ESP32 Cam



Forma finala



Concluzii

Un proiect interesant care m-a scos din zona de confort. Chiar daca nu am reusit sa implementez modulul GSM si GPS, am acumulat multe informatii noi despre cum functioneaza majoritatea lucrurilor din jurul meu la nivel de

hardware. Am invatat sa lipesc piese, sa lucrez cu diverse biblioteci din arduino si cum sa fac debug pe placute Arduino.

Download

Ion Dragos-Cosmin, 336CC → proiect_pm_ion_dragos.zip

Bibliografie/Resurse

1. <https://www.youtube.com/watch?v=UA3cqgpFHXw> [https://www.youtube.com/watch?v=UA3cqgpFHXw]
2. <https://www.youtube.com/watch?v=7-3piBHV1W0&t=71s> [https://www.youtube.com/watch?v=7-3piBHV1W0&t=71s]
3. https://arduinogetstarted.com/tutorials/arduino-neopixel-led-strip?utm_content=cmp-true [https://arduinogetstarted.com/tutorials/arduino-neopixel-led-strip?utm_content=cmp-true]
4. https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/?utm_content=cmp-true [https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/?utm_content=cmp-true]
5. <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/> [https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/]

Link-ul catre pagina

https://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/jewelguard_proximity_defender
[https://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/jewelguard_proximity_defender]

pm/prj2023/apredescu/jewelguard_proximity_defender.txt · Last modified: 2023/05/30 09:15 by dragos_cosmin.ion