

Sistem de alarma

- Student : Şuiu Ana Maria
- Grupa : 332CC

Introducere

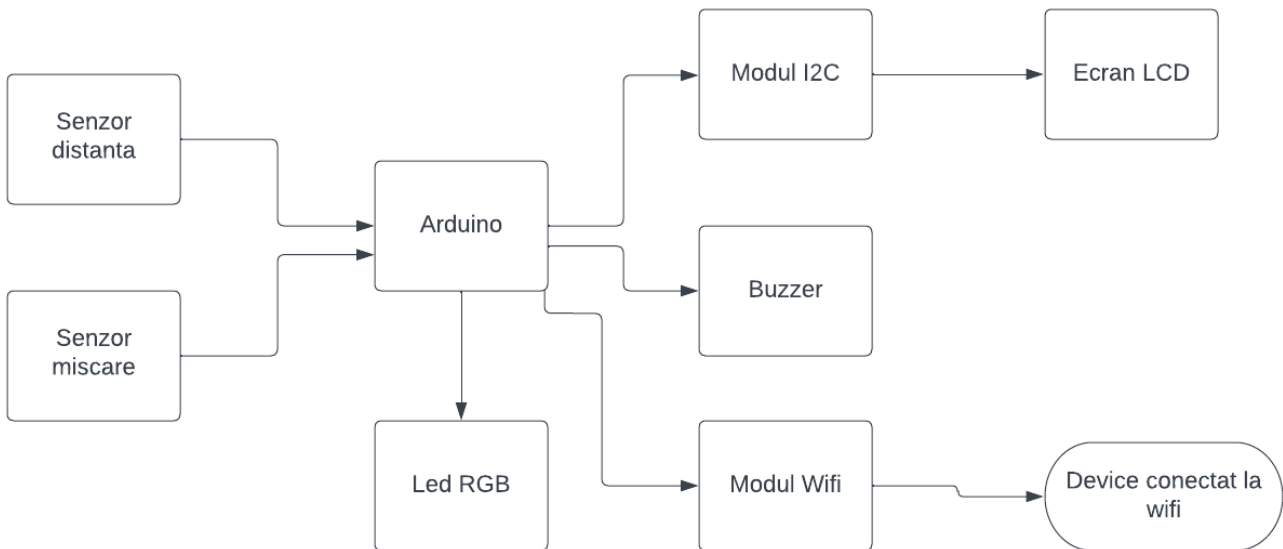
Proiectul consta in implementarea unui sistem de alarma care detecteaza prezenta unei persoane/ unui obiect cu ajutorul unor senzori de distanta si de miscare. Astfel, cel care este conectat prin wifi la sistem va fi instiintat daca s-a detectat o prezenta in apropiere prin culorile date de led, prin pornirea unui alarme si prin primirea unui email.

Scopul proiectului este de a monitoriza prezenta altor obiecte/ persoane straine asupra unui bun, pentru a-l putea tine in siguranta.

Descriere generală


Sistemul functioneaza dupa urmatoarea logica : cand este detectata miscarea unei persoane, este determinata distanta acestuia fata de sistem. Urmeaza sa vedem in ce interval se claseaza distanta respectiva. La o distanta mai mare de 75cm ledul ramane verde. In intervalul [50cm, 75cm] ledul se coloreaza in albastru. In intervalul [25cm, 50cm] ledul se coloreaza in galben si este activata o alarma cu ajutorul unui buzzer. In intervalul [0 cm, 25cm] ledul se coloreaza in rosu, este activata o alarma cu ajutorul buzzerului si persoana conectata la sistem va primi un email. In tot acest timp pe ecranul LCD va fi afisata distanta fata de obiect. Daca nu este detectata prezenta vreunui obiect, pe ecran este afisata un mesaj "Nu s-a detectat miscare!" iar ledul are culoarea verde.

Schema bloc :

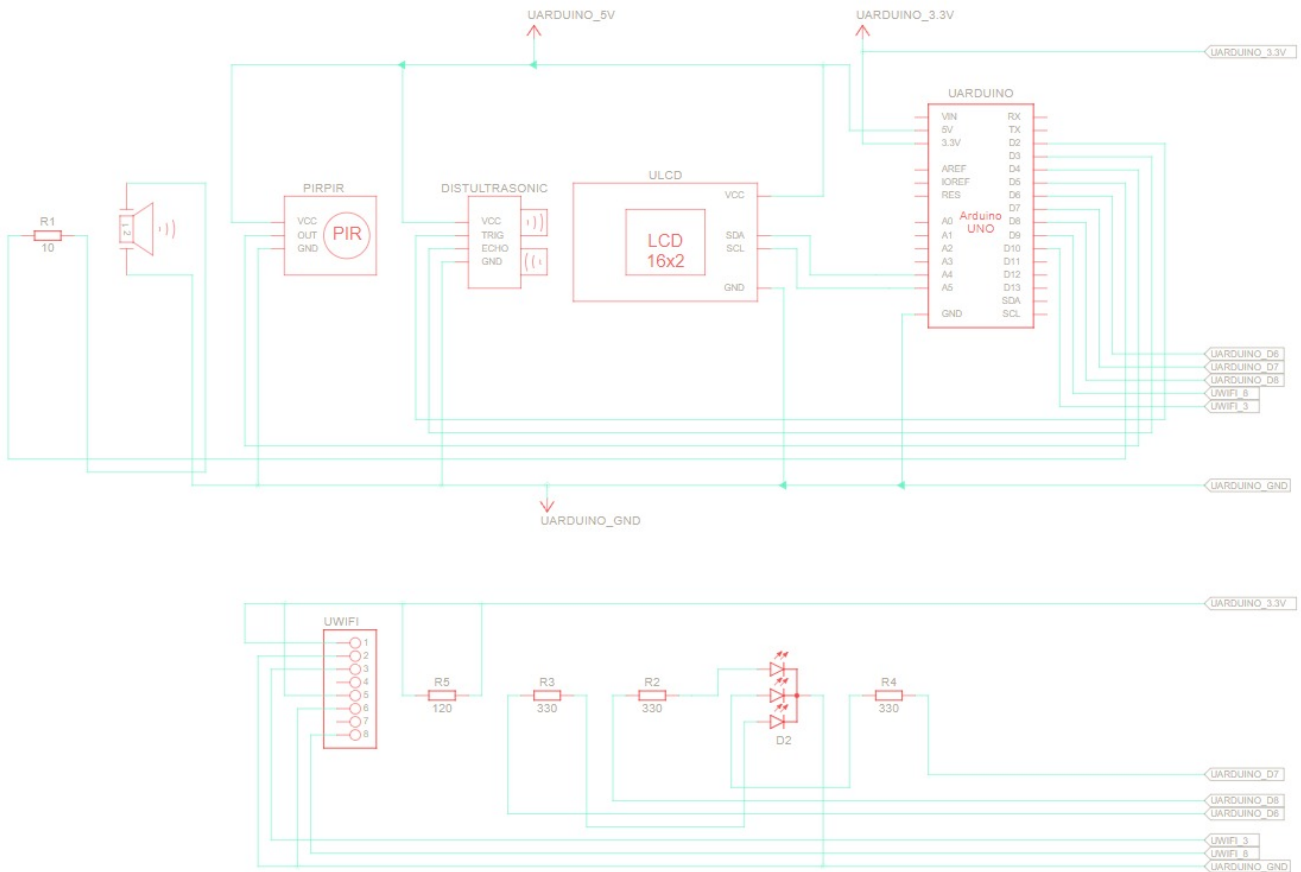
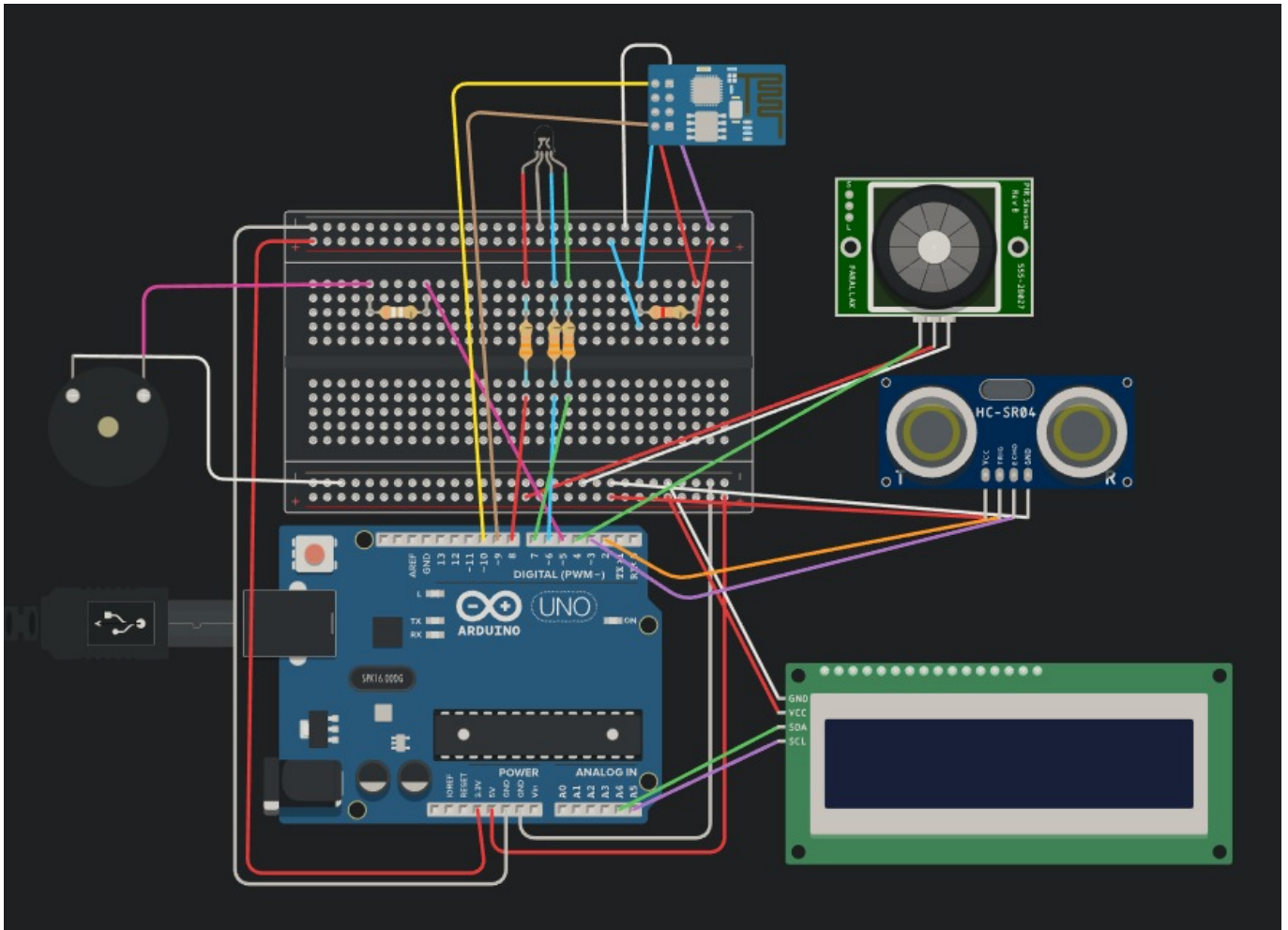


Hardware Design

Lista piese :

- Placa Arduino Uno
- Breadboard
- Senzor PIR HC-SR501
- Senzor ultrasonic HC-SR04
- Led RGB
- Buzzer
- Modul LCD 1602
- Modul I2C PCF8574T
- Modul WI-FI ESP8266
- Rezistente
- Fire
- Cutie, culori, benzi colorate pentru design 

Schema electrica :



Software Design

Mediu de dezvoltare : Arduino IDE

Librării și surse 3rd-party :

- LiquidCrystal_I2C.h
- SoftwareSerial.h
- WiFiEspClient
- <https://ifttt.com/explore>
- <https://ifttt.com/applets/rdyfa36w-if-maker-event-alerta-then-send-yourself-an-email-from-suiuanamaria29-gmail-com>

Implementare software :

La inceputul programului includ bibliotecile necesare pentru utilizarea anumitor componente, acestea fiind "WiFiEsp.h", "SoftwareSerial.h", "LiquidCrystal_I2C.h" si creez cate un obiect pentru fiecare biblioteca. Pentru serial1 apelez folosindu-ma de pinii RX si Tx pentru a configura comunicarea seriala pentru modulul wifi. Realizez initializari pentru modulul wifi, incluzand numele retelei la care ma conectez prin wifi, parola, statusul conectarii la retea, numele serverului unde o sa trimit o cerere pentru a primi un email de alerta.

Realizez asocierea pentru fiecare component cu cate un pin:

```
Digitali
Pin trig (2)
Pin echo (3)
Pin senzor PIR (4)
Buzzer(5)
Pin led B(6)
Pin led G (7)
Pin led R (8)
RX, TX(10, 9)
Analogici
Pin SCL -> A5
Pin SDA -> A4
```

Declar variabile globale ajutatoare.

Functia setup : Configurez LCD-ul. Setez pinii digitali pe INPUT sau OUPUT in functie de caz. Setez baud rate-ul pentru serial si serial1 ca fiind 115200. Incep initializarile pentru wifi.

Functia setup_wifi: Incerc sa ma conectez la retea de wifi folosind credentialele(nume, parola). Fac asta intr-un loop pana cand statusul devine connected si afisez mesaj de succes.

Functia loop: Citesc valoarea senzorului PIR, iar daca aceasta este pe HIGH inseamna ca s-a detectat miscare si setez flag-ul flagDistance pe 1. Daca este pe LOW inseamna ca nu s-a detectat miscare si apelez functia printMessage care afiseaza un mesaj pe LCD "Nu s-a detectat miscare". (Am observat ca nu intra niciodata pe ramura aceasta de else pentru ca senzorul detecteaza mereu miscare).

Daca flagDistance este pe 1, incep sa masor distante dintre senzor si obiect cu ajutorul pinilor Echo si

Trig. Daca distanta este mai mica decat 25cm (aproximativ) astept ca clientul WifiEsp pentru a putea trimite un HTTP request catre server. Apelez functiile toggleLed si printDistance pentru a schimba culoarea ledului in functie de distanta si pentru a printa pe ecranul lcd distanta fata de obiect.

makeHttpRequest(): Incerc sa ma conectez la serverul ifttt pe portul 80. Daca conectare functioneaza, afisez mesaj pe seriala si fac o cerere HTTP de tip POST prin intermediul careia imi va trimite un email de alerta care ma anunta s-a detectat un obiect la mai putin de 25cm fata de sistem.

toggleLed(): In functie de intervalul in care se afla distanta perceputa fata de obiect, fac urmatoarele actiuni astfel: [75cm, ..] → ledul ramane verde

[50cm, 75cm] → ledul se coloreaza in albastru

[25cm, 50cm] → ledul se coloreaza in galben si este apelata functia soundOn() care porneste buzzerul

[0,25cm] → led-ul se coloreaza in rosu si este apelata functia soundOn() care porneste buzzerul (si voi primi si email datorita codului din loop)

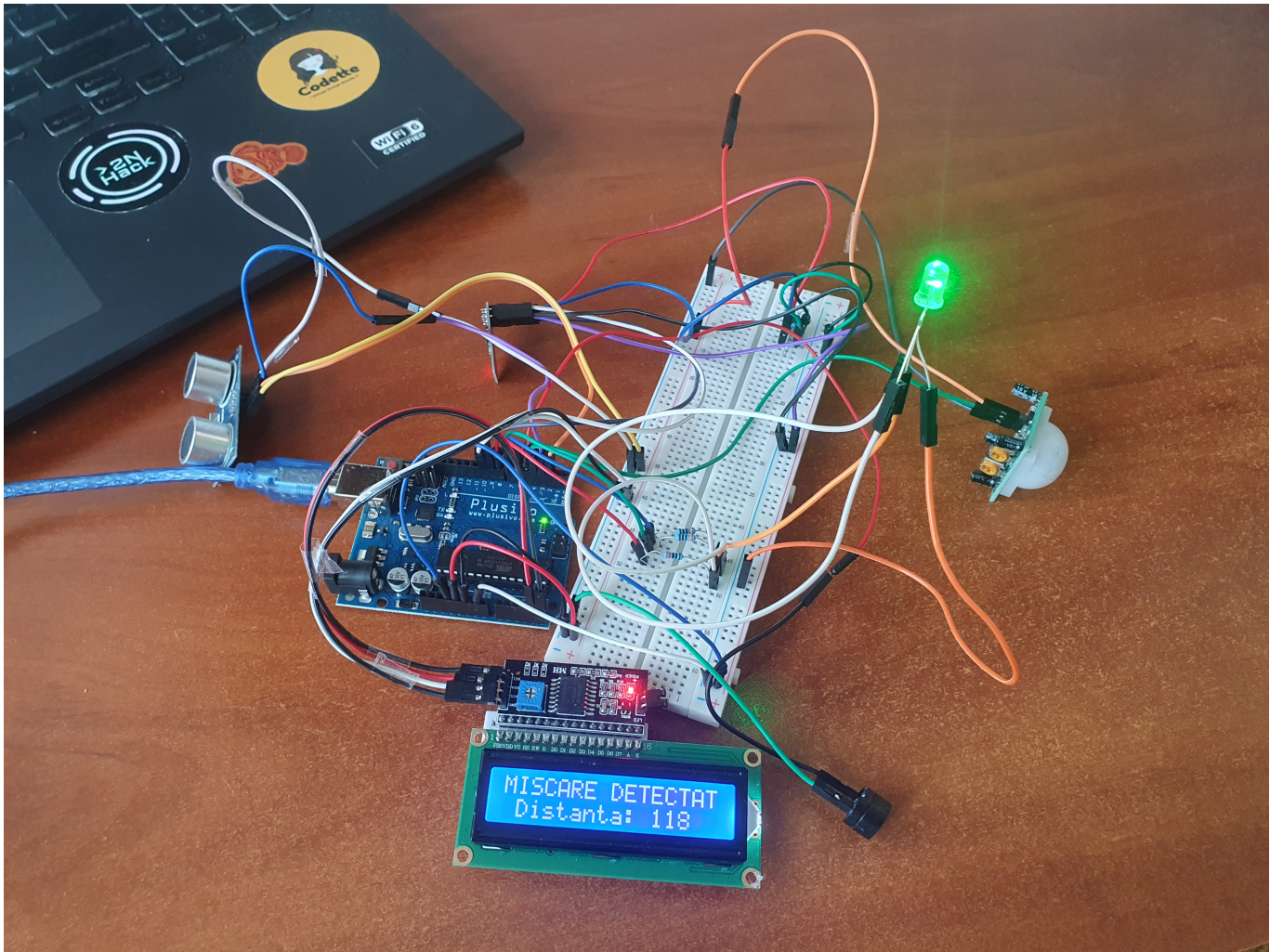
printDistance(): Setez cursorul la inceputul primei linii(0,0) si printez pe acesta "MISCARE DETECTATA!". Setez cursorul la inceputul celei de-a doua linii(1,1) si si afisez distanta perceputa fata de obiect.

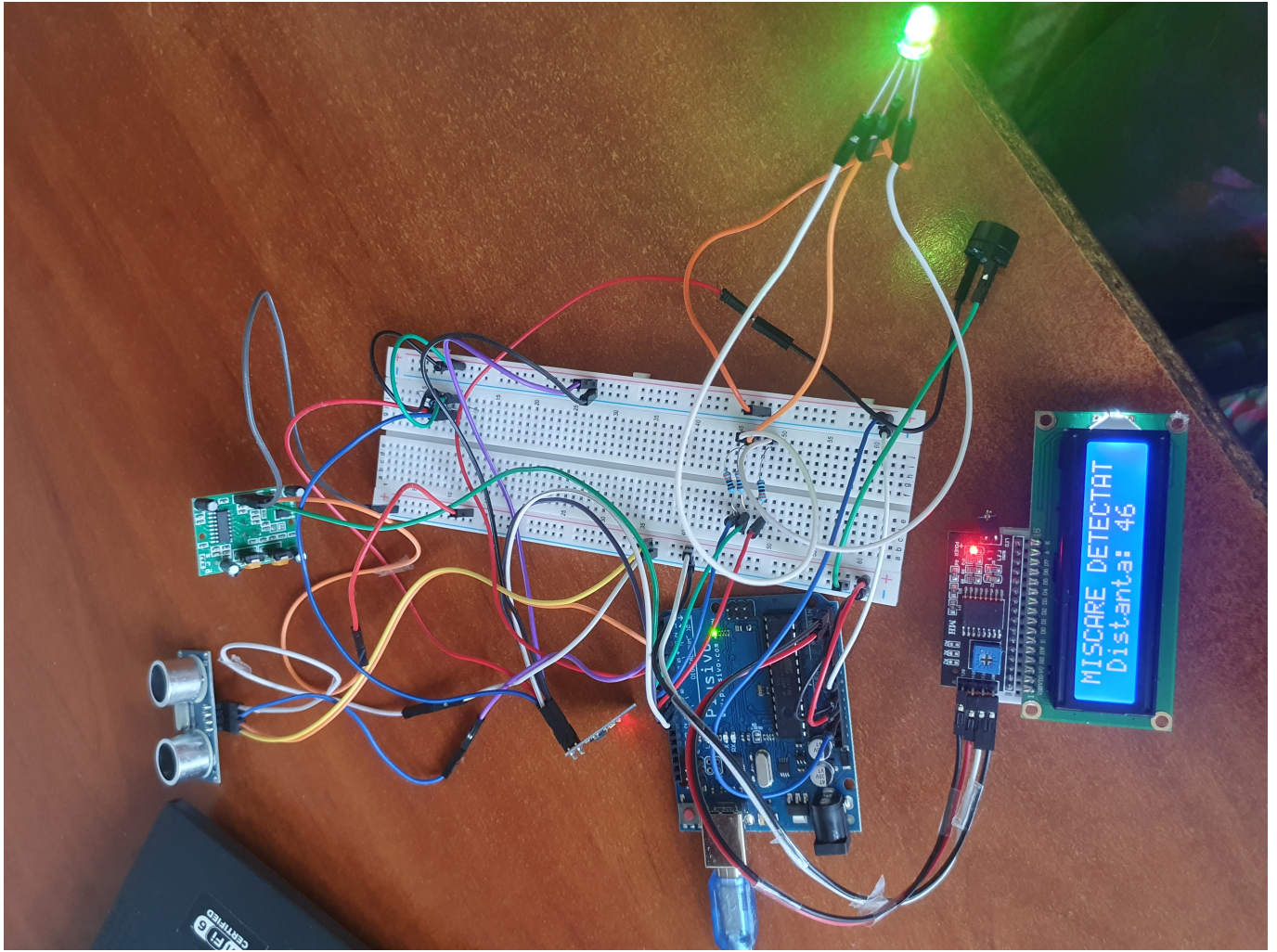
printMessage(): Setez cursorul la inceputul primei linii(0,0) si printez pe acesta "Nu s-a detectat". Setez cursorul la inceputul celei de-a doua linii(1,1) si si afisez " miscare! ".

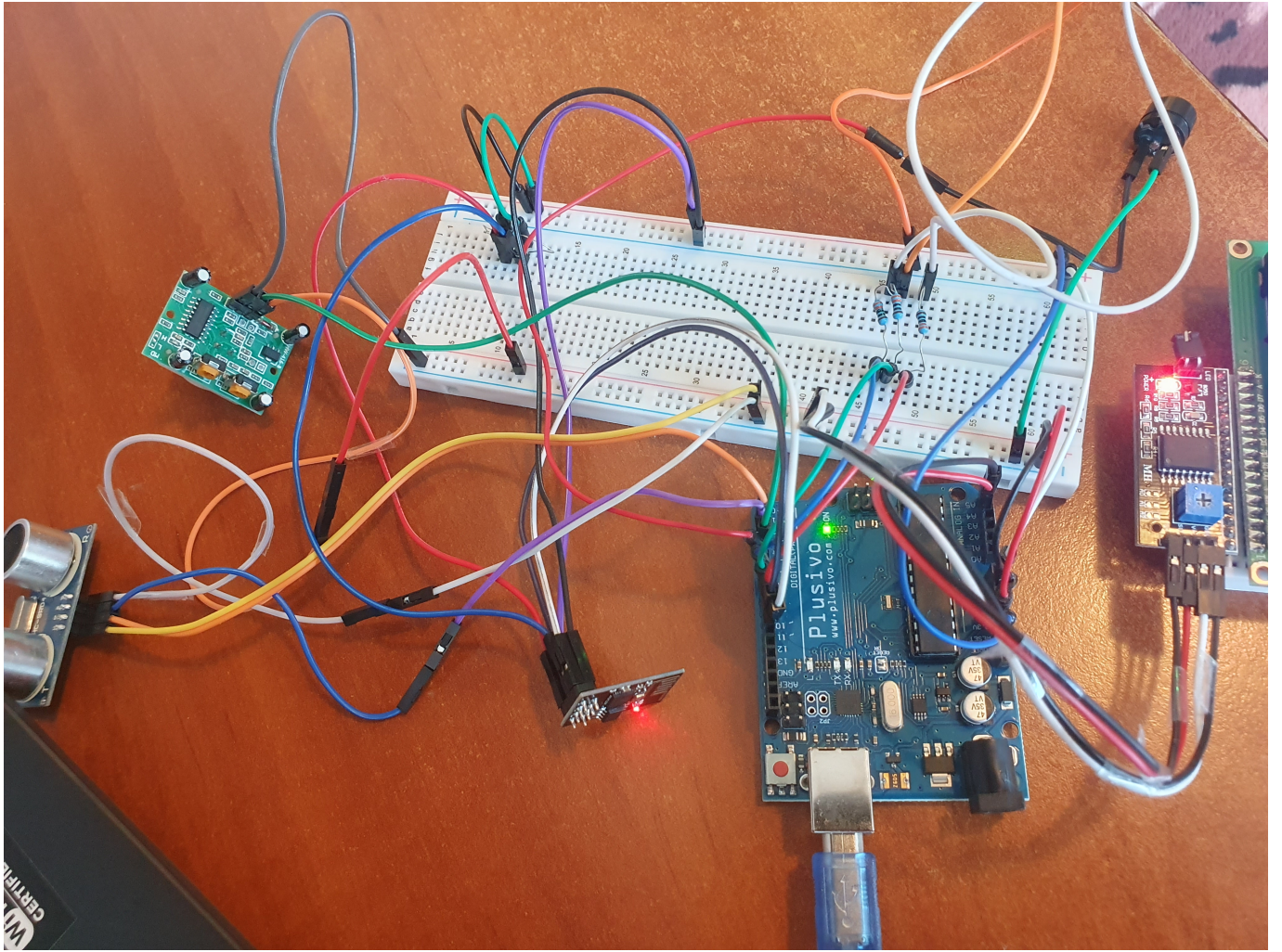
soundOn(): Ma folosesc de functia Tone pentru a seta diferite frecvente pentru a produce un sunet de alarma pe buzzer. Opresc sunetul folosind functia noTone.

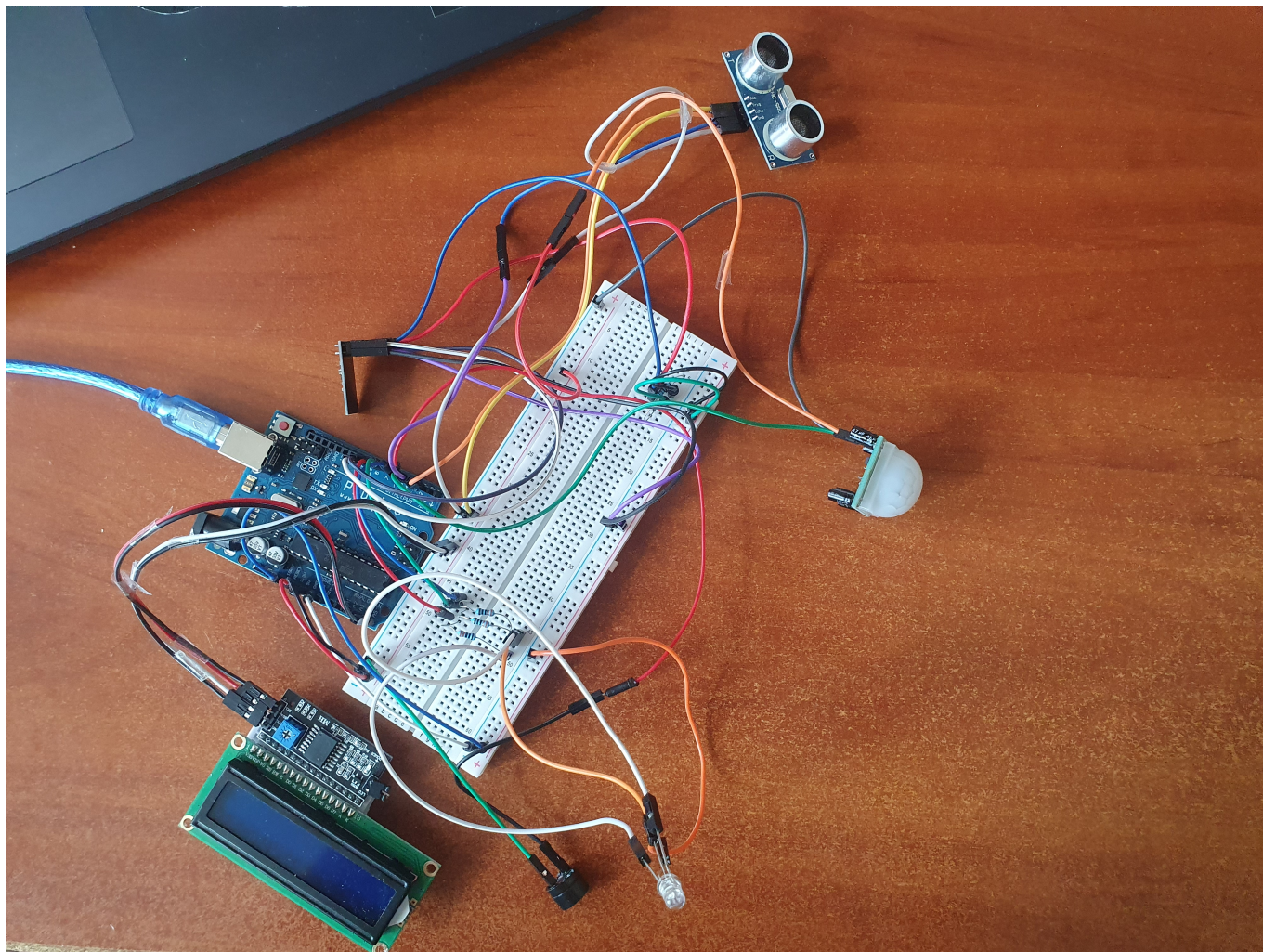
Pentru a transforma din timpul masurat de senzorul ultrasonic in distanta am folosit formula :
 $distance = timeUltrasonic * 0.0343 / 2;$

Rezultate Obținute

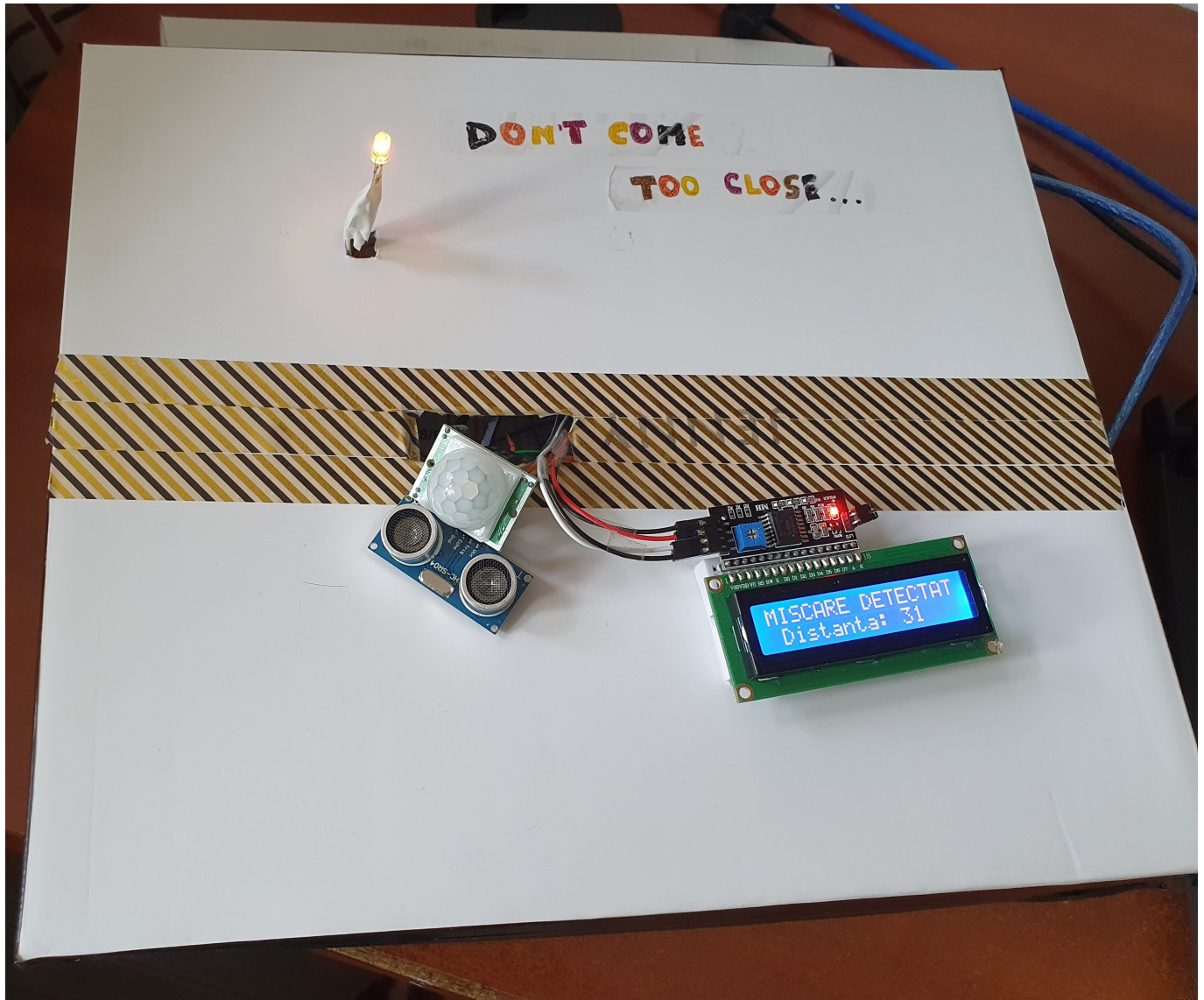








Ulterior, am realizat si acest suport pentru a putea tine senzorii aproape unul de altul si pentru a avea un design mai frumos.



Demo proiect : https://www.youtube.com/watch?v=3XNbpsHQhQ&ab_channel=Ana%C8%98uiu

Concluzii

Per total a fost un proiect interesant si destul de reusit, pot spune ca am invatat destul de multe lucrand la aceasta. Cea mai grea parte a fost lucrul cu modulul wifi deoarece nu il mai folosisem pana acum si mi-a luat ceva timp pana sa-mi dau seama cum functioneaza. Si in plus nu mergea intotdeauna conectarea la wifi si conectarea la server pentru a primi email. ❌

Download

[sistem_de_alarma.zip](#)

Jurnal

25.04.2023 : Alegere tema proiectului

1.05 - 5.05.2023 : Am ales ce piese folosesc si am decis logica implementarii

8.05 - 14.05.2023 : Am ansamblat piesele si terminat partea hardware

15.05. - 21.05.2023 : Am inceput implementarea software

22.05 - 26.05.2023 : Am terminat partea software

27.05. - 29.05.2023 : Ultimele retusuri + realizat designul sistemului + finalizat documentatia

Bibliografie/Resurse

<https://github.com/bportaluri/WiFiEsp/tree/master/examples/WebClient>

<https://docs.arduino.cc/learn/built-in-libraries/software-serial>

<https://www.elementzonline.com/blog/ultrasonic-sensor-hc-sr04-with-arduino>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/dene/sistemdealarma>



Last update: **2023/05/30 12:21**