

**PROIECT INFORMATICĂ APLICATĂ II**

**Coordonator:**

**Profesor: Oproiu Mihai**

**Studentă:**

**Halmaghi Alexandra-Maria**

**Grupa.4LF613 (A)**

**2022-2023**

**1.De ce am ales acest proiect?**

În urma diferitelor dificultăți întâmpinate în parcarea autovehiculului, cauzate de locurile de parcare strâmte și îngrămădite, m-am simțit determinată să îmi instalez un senzor pentru a-mi ușura munca în trafic. Costurile implementării, și in special a senzorului, mi-au stârnit diferite curiozități, mai exact: de ce valorează atât, dar și cât costă confecționarea lor. Însă, dorința mea de cunoaștere nu s-a oprit în acel punct, următoarele întrebări fiind: care este principiu de funcționare al acestui senzor, cum și cat de greu se confecționează. Ca să îmi răspund la întrebări, m-am gândit ca cel mai potrivit mod este de a crea un prototip. Si, de ce nu, pe viitor să încerc să îl montez chiar eu pe propriul vehicul.

**2.Care este utilitatea?**

Senzorii sunt folosiți ca ajutor suplimentar atunci când parchezi. Acești senzori folosesc sunete de frecvență înaltă pentru a detecta obiectele.

Aceste sunete creează o undă care este reflectată de obiectele din jur. Un receiver(receptor) primește sunetele reflectate de obiecte și calculează distanța până la ele.

Această tehnică a fost inspirată din natură, de la lilieci, care folosesc același sistem pentru a detecta obiectele din jur.

**3.Cum a fost realizat proiectul?**

**3.1 Componente hardware.**

În primul rând am achiziționat componentele /materialele necesare pentru asamblarea lucrării:

-Placă de dezvoltare (ARDUINO) UNOR3

-Buzzer 5V

-Breadboard

-Senzor cu ultrasunet HC-SR04

-10 Fire Dupont tată – tată

-LED roșu 5mm

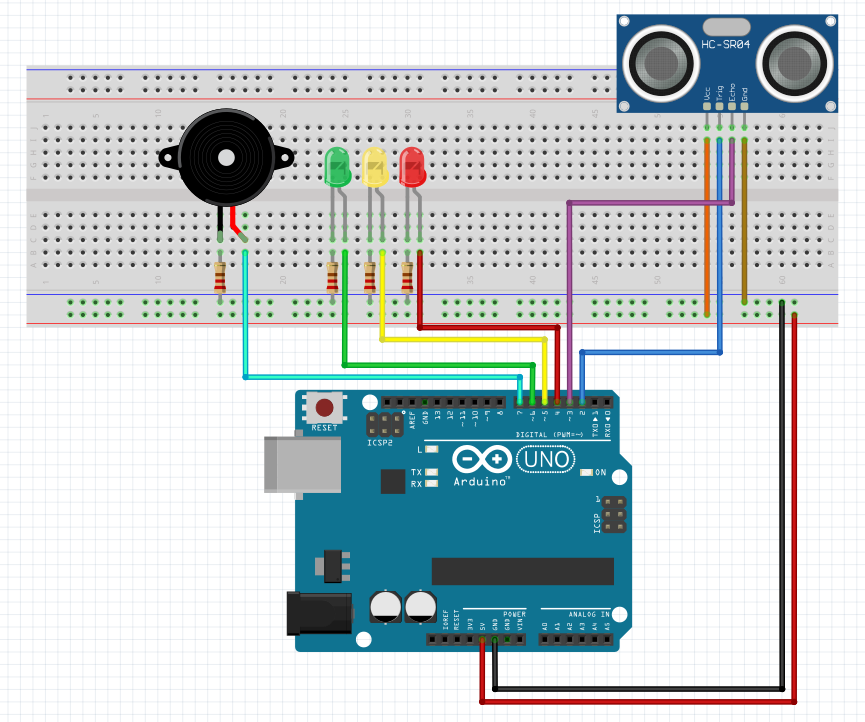
-LED verde 5mm

-Led galben 5mm

-4xRezistor de 220Ω

-Cablu USB pentru UNO

**3.2 Configurare/Interfață în aplicție (Fritzin)**

****

**3.3 Asamblare**

* Am conectat un fir roșu la pinul de 5V de pe Arduino la canalul pozitiv al plăcii și un fir negru de la pinul GND la canalul negativ al plăcii.
* Am plasat senzorul cu ultrasunete HC-SRO4 în așa fel încât să fie orientat spre exterior, am conectat pinul GND al senzorului la canalul negatic al plăcii,apoi TRIG de pe senzor la pinul 2 de pe Arduino , pinul ECHO de pe senzor la pinul 3 iar VCC la canalul pozitiv de pe placă.
* Următorul pas a constat ăn conectarea LED-urilor .

LED-ul verde a fost coectat de la anod(piciorul mai lung) la pinul 6 de pe Ardino cu un fir verde, iar catodul(piciorul mai scurt) la canalul negativ de pe placă folosint un REZISTOR DE 220Ω.

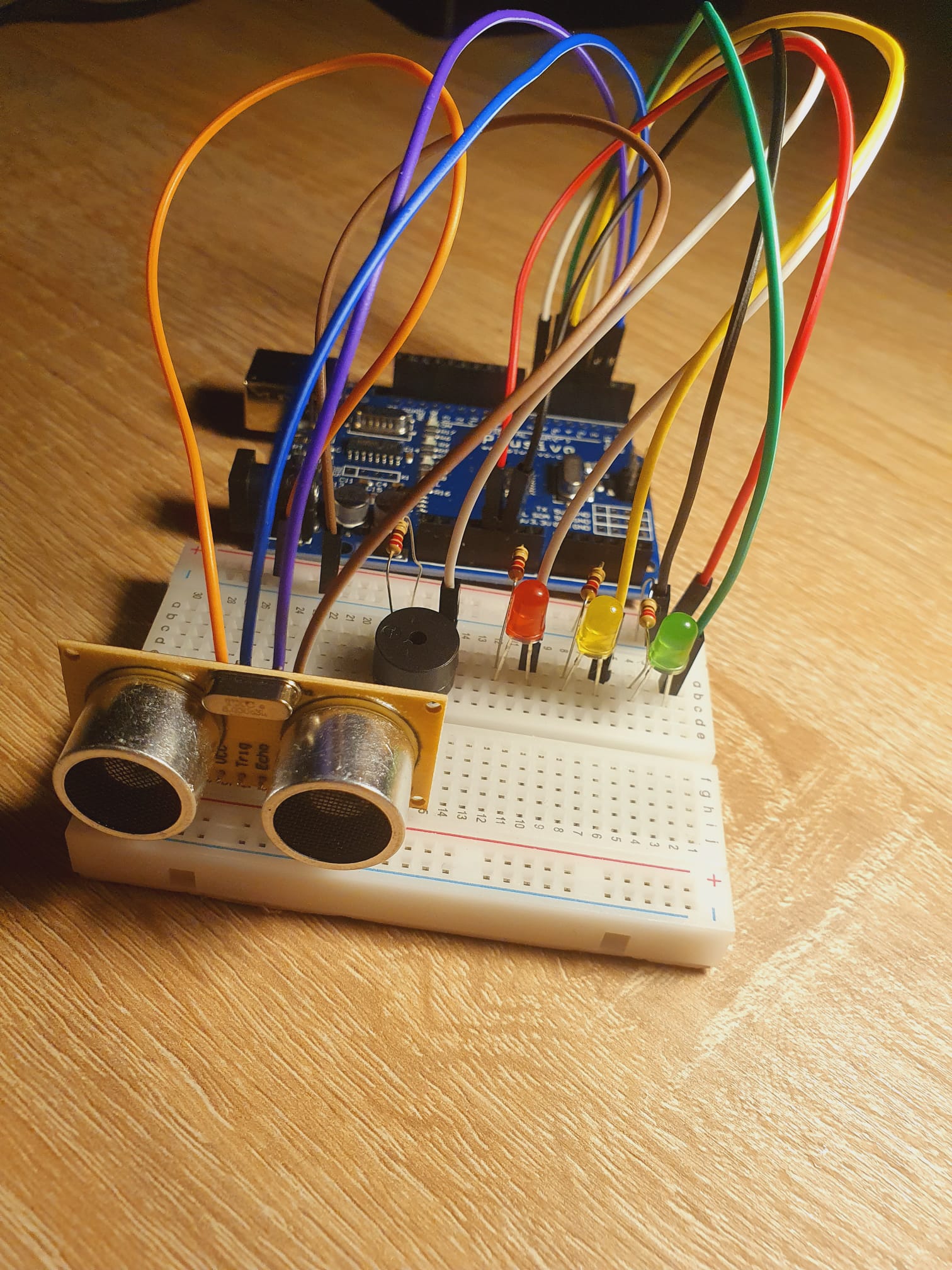
Analog pentru LED-ul roșu (l-am conectat la pinul 4) și LED-ul galben(pin 5)

* Ultima asamblare este conectarea soneriei la placa de breadboard și Arduino.

Am conectat piciorul mai lung la pinul 7 folosid un fir dupont tată-tată, iar piciorul mai scurt al soneriei la canalul negativ al plăcii de breadboard folosind un rezistor 220Ω pentru a reduce volumul soneriei.

**3.4 Finalizare asamblare**

Imagine personală



**3.5 Cod sursă**

#define trigPin 2

#define echoPin 3

#define LEDlampRed 4

#define LEDlampYellow 5

#define LEDlampGreen 6

#define soundbuzzer 7

int sound = 500;

void setup() {

Serial.begin (9600);

//viteză de transmitere/ Pornește comunicarea serială

pinMode(trigPin, OUTPUT);

//setez pinul ca ieșire

pinMode(echoPin, INPUT);

pinMode(LEDlampRed, OUTPUT);

pinMode(LEDlampYellow, OUTPUT);

pinMode(LEDlampGreen, OUTPUT);

pinMode(soundbuzzer, OUTPUT);

}

void loop() {

long durationindigit, distanceincm;

digitalWrite(trigPin, LOW);// Șterge întârzierea

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(15);//setează trigPin pe starea HIGH pentru 10 microsecunde

digitalWrite(trigPin, LOW);//unda sonoră se va termina

durationindigit = pulseIn(echoPin, HIGH);// Citește echoPin, returnează timpul de călătorie a undei sonore în microsecunde

distanceincm = (durationindigit/5) / 29.1;// Calcularea distanței

if (distanceincm < 50) {

digitalWrite(LEDlampGreen, HIGH);

}

else {

digitalWrite(LEDlampGreen, LOW);

}

if (distanceincm < 20) {

digitalWrite(LEDlampYellow, HIGH);

}

else {

digitalWrite(LEDlampYellow,LOW);

}

if (distanceincm < 5) {

digitalWrite(LEDlampRed, HIGH);

sound = 1000;

}

else {

digitalWrite(LEDlampRed,LOW);

}

if (distanceincm > 5 || distanceincm <= 0){

Serial.println("În afara intervalului permis de distanțe");

noTone(soundbuzzer);

}

else {

Serial.print(distanceincm);// Tipărește distanța pe monitor

Serial.println(" cm");

tone(soundbuzzer, sound);

}

delay(300);

}

**4.Bibliografie.**

* <https://arduinolearn.github.io/ultra.html>
* <https://www.youtube.com/watch?v=-H6zhUVFAs0>
* <https://www.instructables.com/Arduino-Distance-Detector-with-a-Buzzer-and-LEDs/>
* <https://projecthub.arduino.cc/viniciuslindoss/bcf4d6ec-375e-49fa-8293-e3eceb9f2f96?ref=search&ref_id=parking%20senzor&offset=0>
* <https://fritzing.org/download/> (aplicație pentru interfață)