

# Proiectare cu microprocesoare

## -- Senzori de parcare –



**STUDENTI:**

**BUMBU RAZVAN**

**FLOREAN EUSEBIU**

**PROF. INDRUMATOR:**

**ITU RAZVAN**

CUPRINS:

1. Tema abordata
2. Solutia problemei
3. Diagrama Circuitului
4. Bibliografie

## 1. Tema abordata

Tema abordata in cadrul acestui proiect este reprezentata de crearea unui sistem de senzori de parcare. Senzorii de parcare sunt niște dispozitive electronice de proximitate, care detectează obiectele sau persoanele ce se află în imediata apropiere a mașinii.

Senzorii de parcare au fost inventati de catre Tony Heyes. Acesta avea el însuși probleme cu un ochi, cu care nu vedea deloc. Acesta s-a gândit că ar fi ideal un sistem similar pentru mașini, motiv pentru care el l-a adaptat pe propriul autovehicul. In 1985, Heyes s-a dus la cei de la Jaguar pentru a le vinde aceasta inventie, însă cei de acolo l-au refuzat politicos spunându-i că sistemul este folositor unei persoane cu un singur ochi valid, care nu poate aprecia distanța și că persoanele sănătoase nu vor dori un astfel de sistem.



In 1993, Ford Germania lanseaza pe piata primul sistem de asistenta la parcare Bosch, ca dotare optionala pentru modelul Ford Scorpio.



Spre deosebire de perioada prezentata mai sus, in prezent senzorii de parcare sunt foarte apreciati, aducand o serie de avantaje, ajucand conducatorul auto sa parcheze in siguranta. Acestia nu sunt instalati doar pentru a asigura masina, ci si pe cei din jurul acesteia.

Senzorii de parcare sunt niște dispozitive electronice de proximitate, care detectează obiectele sau persoanele ce se află în imediata apropiere a mașinii. Functionarea acestora, conform proiectului realizat, constau in mai multi senzori ultrasonici care masoara distanta, iar in functie de aceasta se realizeaza un sunet a carui intensitate creste cu scaderea distantei. Pentru a facilita utilizarea senzorilor, masuratorile efectuate de catre acestia sunt illustrate cu ajutorul unei interfete grafice foarte sugestive.

## 2.Solutia problemei

Pentru implementarea proiectului am folosit ca medii de programare Arduino prin care am programat placuta si Processing cu ajutorul caruia am realizat interfata grafica. Ca si componente hardware, am folosit:

- Placa de dezvoltare Arduino Uno  
-aceasta a fost programata cu ajutorul mediului de programare Arduino
- Doi senzori ultrasonici  
-acestia prezinta 2 componente, prin una dintre acestea este transmis sunetul, iar cealalta componenta il receptioneaza, reusind astfel sa calculeze distanta in functie de timpul necesar receptionarii sunetului emis.
- Doua buzzere  
-au fost folosite pentru a justifica functionarea corecta a senzorilor, emitand sunete care depend invers proportional de distanta pana la obstacol.
- Breadboard  
-folosita pentru o mai buna conectare a componentelor, astfel evitand utilizarea unui numar mare de fire
- fire de legatura tip tata-tata  
-conectarea componentelor

### ARDUINO:

In cadrul functiei setup() are loc initializarea pinilor ( intrare/iesire). Intrucat avem 2 senzori ultrasonici si 2 buzzere, a fost nevoie de 2 pini de intrare(echoPinRear si echoPinFront ) si de 4 pini de iesire(trigPinRear, trigPinFront, buzzerRear, buzzerFront). De asemenea, tot in cadrul acestei functii are loc initializarea frecventei microcontrollerului.

In cadrul functiei loop, are loc apelarea functiilor care pornesc buzzerele in functie de distanta, masoara distanta pana la obstacol si afisarea in interfata seriala a datelor masurate de catre acestea.

```
void setup(){  
  pinMode(trigPinRear, OUTPUT);  
  pinMode(echoPinRear, INPUT);  
  pinMode(trigPinFront, OUTPUT);  
  pinMode(echoPinFront, INPUT);  
  pinMode(buzzerRear, OUTPUT);  
  pinMode(buzzerFront, OUTPUT);  
  Serial.begin(9600);  
}
```

In cadrul acestei functii are loc calcularea distantei in functie de datele masurate de catre senzorul ultrasonic din fata automobilului, similar celui din spatele acestuia. Prin functia digitalWrite are loc scrierea in pinul de iesire 1,0 si 1, ceea ce declanseaza masurarea. In variabila durationRear se mentine timpul masurat intre emiterea-receptionarea sunetului, iar in distanceRear are loc convertirea acestuia in distanta. Aceasta convertire este realizata prin inmultirea timpului cu viteza sunetului. La final se imparte rezultatul la 2 intrucat interesul este de a masura distanta de la senzor pana la obstacol.(fiind neglijata distanta de la obstacol inapoi la senzor).

In cazul functiei alarmRear() se stimuleaza sunetul buzzerul corespunzator senzorului. Aceasta functie controleaza atat intensitatea sunetului, cat si frecventa acestuia. Pentru a ilustra frecventa sunetului, am introdus un delay dupa emiterea sunetului care scade o data cu scaderea distantei, astfel frecventa sunetului va creste cu cat ne apropiem mai tare de obstacol(la fel si intensitatea lui). Pentru a seta intensitatea sunetului am folosit functia tone().

```
void measureRearDist () {  
    digitalWrite(trigPinRear, LOW);  
    digitalWrite(trigPinRear, HIGH);  
    digitalWrite(trigPinRear, LOW);  
    durationRear = pulseIn(echoPinRear, HIGH);  
    distanceRear = durationRear * 0.034 / 2 / 3;  
}  
  
void alarmRear() {  
    if (distanceRear >= 9){  
        noTone(buzzerRear);  
    }  
    else if (distanceRear >= 6 && distanceRear < 9){  
        tone(buzzerRear, 261);  
        delay(150);  
        noTone(buzzerRear);  
    }  
    else if (distanceRear >= 3 && distanceRear < 6){  
        tone(buzzerRear, 330);  
        delay(100);  
        noTone(buzzerRear);  
    }  
    else if (distanceRear > 0 && distanceRear < 3){  
        tone(buzzerRear, 440);  
        delay(50);  
        noTone(buzzerRear);  
    }  
    else if (distanceRear == 0){  
        tone(buzzerRear, 440);  
    }  
}
```

## PROCESSING:

Acest mediu de programare a fost folosit pentru crearea interfetei grafice. Acesta este similar Arduino, astfel ca prezinta o functie de setup() si o functie de draw(). Avand declarant un obiect al clasei Serial, vom putea sa citim datele din interfata seriala din Arduino si sa le utilizam pentru a crea o imagine dinamica care sa reactioneze in functie de distanta dintre senzor si obstacol.

In cadrul functiei setup(), are loc setarea fundalului si desenarea celor 6 forme care se activeaza in momentul cand distanta scade sub o anumita limita. In functie de distanta, aceste forme vor primi o culoare(verde, galben sau rosu) . Prin intermediul functiei fill() am dat codul de culoare in format rgb(), prin stroke am specificat culoarea marginilor iar prin rect() are loc desenarea dreptunghiurilor la coordonatele specificate prin intermediul parametrilor.

```
----- \-----, -----,  
bg = loadImage("G30.jpg");  
background(bg);  
fill(56, 56, 56);  
stroke(56, 56, 56);  
rect(1600, 120, 45, 300, 5, 100, 100, 5);  
----- ,-----, -----
```

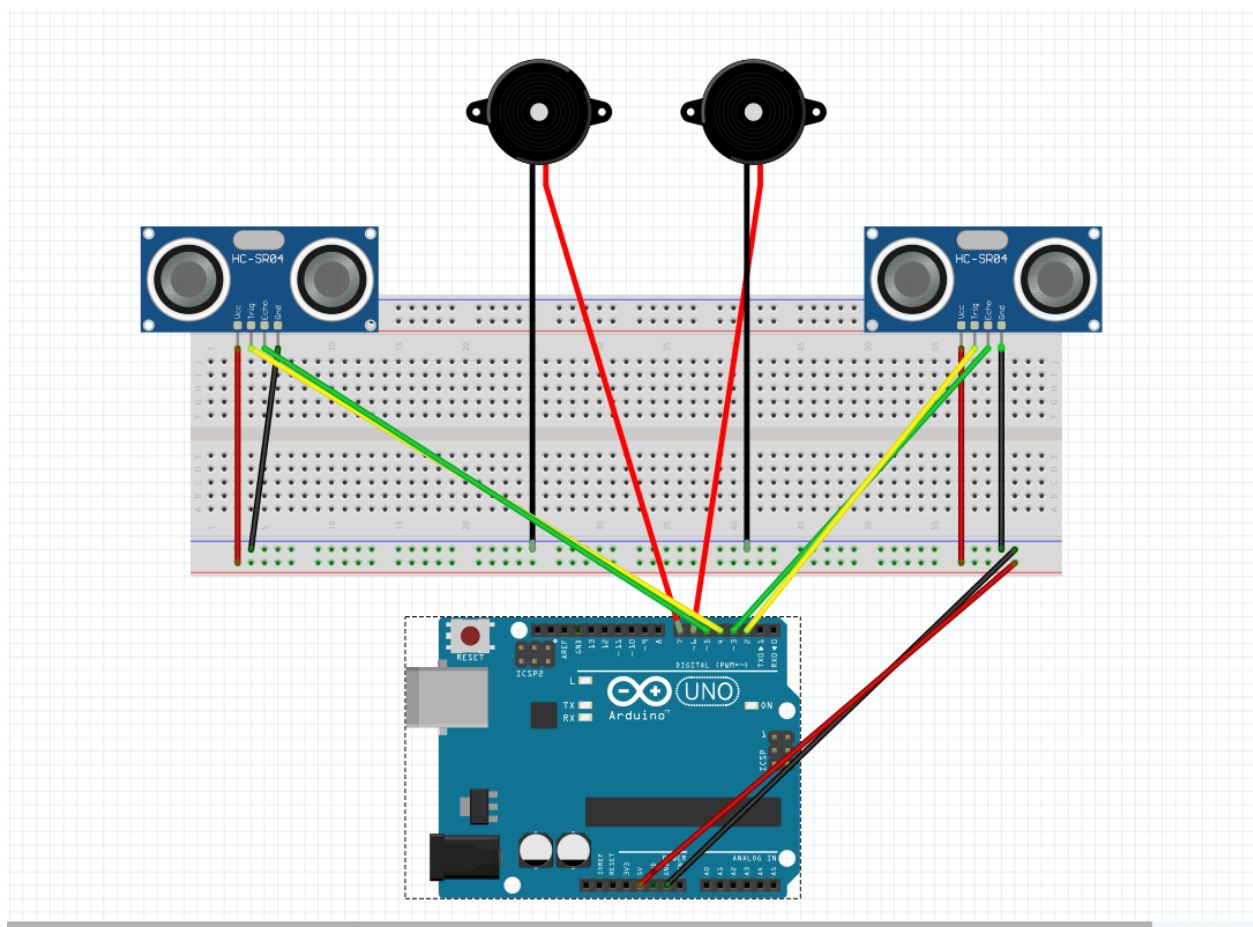
In functia draw, are loc redesenarea patratelor, in functie de datele citite din interfata seriala. Pentru a interpreta datele, am citit intregul mesaj ca un string pe care l-am divizat dupa caracterul "\n", memorand doar datele de interes( distanceRear, distanceFront). Aceasta divizare a fost realizata cu ajutorul functiei split(). Pentru a verifica cand sunt date disponibile pentru a fi citite din interfata seriala, am utilizat functia Serial.available().

```
void draw() {  
  if (mySerial.available() > 0) {  
    readString = mySerial.readString();  
  
    println("-----\n" + readString);  
  
    if (readString != null){  
      String[] strings = readString.split("\n", 4);  
  
      if (strings[0].contains("REAR")){  
        distanceRear = float(strings[1]);  
      }  
  
      if (strings[2].contains("FRONT")){  
        distanceFront = float(strings[3]);  
      }  
    }  
  }  
}
```

La fiecare executie a acestei functii are loc desenarea dreptunghiurilor in culoarea fundalului, ceea ce reprezinta reinitializarea ferestrei. Apoi prin intermediul unei secvente if else se selecteaza cazul in care suntem in functie de valorile inregistrate de senzori, actualizand fereastra prin desenarea dreptunghiurilor corespunzatoare in culoarea potrivita. Pentru selectarea dreptunghiurilor care trebuie desenate, am memorat in variabilele.

### 3. Diagrama Circuitului

Aceasta diagrama a fost realizata cu ajutorul aplicatiei Fritzing si ilustreaza modul in care au fost conectate toate componentele intre ele, inclusive componentele hardware utilizate.



#### 4. Bibliografie

- [https://www.youtube.com/watch?v=ZejQOX69K5M&ab\\_channel=HowToMechatronics](https://www.youtube.com/watch?v=ZejQOX69K5M&ab_channel=HowToMechatronics)
- [https://www.youtube.com/watch?v=NhyB00J6PiM&ab\\_channel=ProgrammingforPeople](https://www.youtube.com/watch?v=NhyB00J6PiM&ab_channel=ProgrammingforPeople)
- [https://www.youtube.com/watch?v=K8AnlUT0ng0&ab\\_channel=Sam%27sNeatProjects](https://www.youtube.com/watch?v=K8AnlUT0ng0&ab_channel=Sam%27sNeatProjects)
- <https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/336-3.pdf>
- [https://docs.google.com/document/d/17wzHexxRoio\\_OoiLF8cciTCVuz9IOU1U2KdN4\\_gWxnl/edit](https://docs.google.com/document/d/17wzHexxRoio_OoiLF8cciTCVuz9IOU1U2KdN4_gWxnl/edit)