



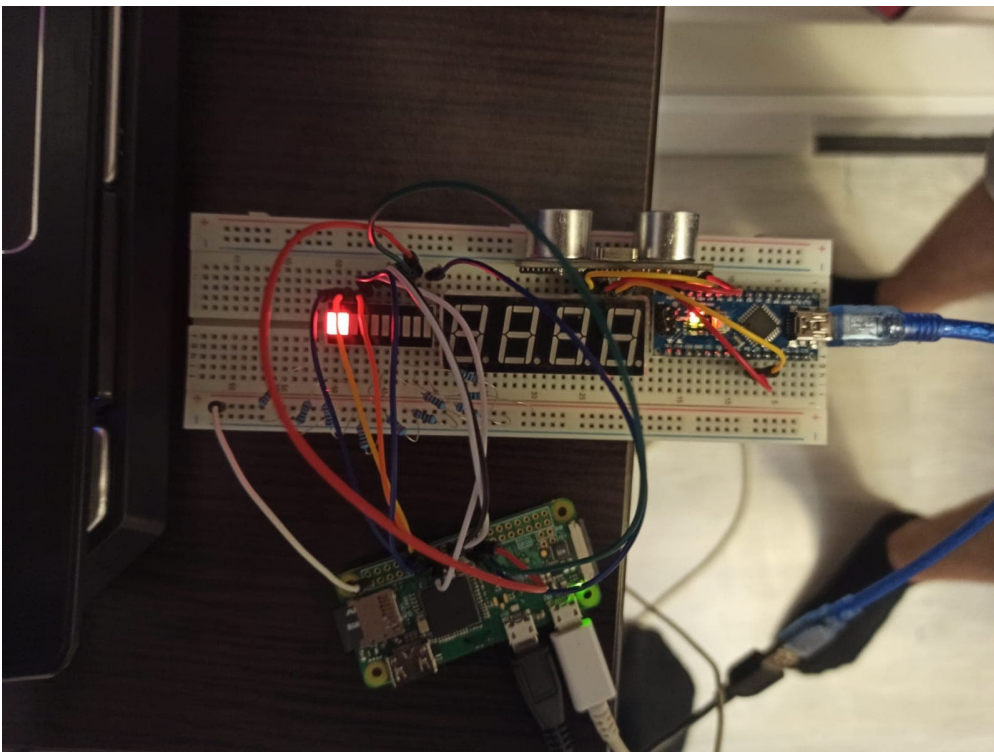
**Simion Emilian**

[emilian.simion1@gmail.com](mailto:emilian.simion1@gmail.com)

**Popescu George**

[popescu.george2304@gmail.com](mailto:popescu.george2304@gmail.com)

**Titlul proiectului:** Senzor de parcare folosind HC sr04 cu notificare prin email.



**Descriere:** Folosim Arduino nano pentru a citi distanta pana la primul obiect, trimitem pe seriala catre raspbery pi zero w, dupa in functie de distanta aprindem o bara de leduri. Cand distanta pana la primul obiect este mai mica de 10 cm, o perioada mai mare de 3 secunde atunci trimitem un email cu o atentionare. Acest senzor poate fi folosit si ca un dispozitiv antifurt.

**Poveste:** Senzorul este folosit pentru o parcare sigura, astfel aprindem o bara de leduri in functie de distanta pana la primul obiect. Cand nu esti in masina acest dispozitiv te atentioneaza cu o notificare prin email daca cineva se aproprie de masina ta. Acest cod functioneaza doar cu o adresa gmail.

#### **Cum functioneaza ?**

- Arduino nano citeste distanta pana la primul obiect folosind un senzor HC sr04, si o trimite pe seriala catre raspbery pi.
- Bara este formata din 10 leduri. Daca distanta este mai mica de 10 cm, atunci toate ledurile

se vor aprinde. In functie de distanta vom aprinde un numar de leduri. Masura fiind din 10 in 10 cm.

- Daca un obiect este la o distanta mai mica de 10 cm fata de senzor, un timp de cel putin 3 secunde, atunci un email va fi trimis catre adresa de email specificata.

#### **Componente Hardware:**

- Raspberry pi zero w
- Arduino nano
- Senzor HC SR04
- Breadboard
- Bara cu 10 leduri
- Fire mama tata x10
- Fire tata tata x5
- Rezistente de 220 Ω x10
- Cablu USB OTG

#### **Aplicatii:**

- Raspberry Pi Raspbian lite
- Arduino Genuino

#### **Cod:**

##### **Arduino**

```
#define echoPin 7
      #define trigPin 8

      long duration;
      long distance;

      void setup() {
        Serial.begin(9600);
        pinMode(echoPin, INPUT);
        pinMode(trigPin, OUTPUT);
      }
      void loop() {
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);

        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);

        digitalWrite(trigPin, LOW);
        duration = pulseIn(echoPin, HIGH);
```

```

distance = duration / 58.2;
Serial.println(distance);
delay(500);
}

```

Explicatie: Motivul pentru care am impartit la 58.2 este pentru ca viteza sunetului este 340m/s. Noi masuram timpul de cand a plecat unda sonora din trigger pana cand ajunge in echo, dar avem nevoie doar de o parte, deci impartim la 2. Avem 170m/s, 17000cm/s  
Distanta = 17000cm/s\*durata, unde durata este masurata in microsecunde, deci transformam in secunde  
Distanta = durata \*17/1000 = durata/1000/17 = durata /58.82 cm

### Raspberry pi

```

import serial
import RPi.GPIO as GPIO

ser = serial.Serial('/dev/ttyUSB0')
ser.flushInput
GPIO.setwarnings(False)

ledPins = [11, 12, 13, 15, 16, 18, 19, 21, 22, 23]
def setup():
    print("Program is starting")
    GPIO.setmode(GPIO.BOARD) #physical location
    for pin in ledPins:
        GPIO.setup(pin, GPIO.OUT) #mode output
        GPIO.output(pin, GPIO.HIGH) #off led

def destroy():
    for pin in ledPins:
        GPIO.output(pin, GPIO.HIGH)
        GPIO.cleanup()

if __name__ == '__main__':
    dict = {"5": "9", "10": "8", "15": "7", "20": "6", "25": "5", "30": "4", "35": "3", "40": "2", "50": "1", "60": "0" }
    dict = {int(k):int(i) for k, i in dict.items()}
    list = dict.keys()
    list.sort()
    numarLeduri = 0

```

```

flag = 0

setup()
try:
    while True:
        ser_bytes = ser.readline()
        decoded_bytes = float(ser_bytes[0:len(ser_bytes)-2].decode("utf-8"))
        dist = decoded_bytes

        ##
        lungime = len(list)
        for i in range(lungime - 1):
            if dist >= list[i] and dist < list[i+1]:
                numarLeduri = dict[list[i]]
                continue
            if dist >= 60:
                numarLeduri = 0

        for pin in ledPins[:numarLeduri]:
            GPIO.output(pin, GPIO.LOW)
        for pin in ledPins[numarLeduri+1:]:
            GPIO.output(pin, GPIO.HIGH)
        ##
        if dist<=10:
            flag = flag + 1
            if flag == 6:
                import smtplib
                server=smtplib.SMTP('smtp.gmail.com',587)

                server.starttls()

                server.login("emiraspberrypi0","parola")
                msg="Alerta senzor, distana este " +
str(dist)

                server.sendmail("emiraspberrypi0@gmail.com",
"naimebale@yahoo.com",msg)
                server.quit()
        else:

```

```

flag = 0

        ##
        print("distanta este {}, iar noi ar trebui sa aprindem primele {}
becuri".format(dist, numarLeduri))
    except KeyboardInterrupt:
        #destroy()
        print("exit")

```

### **Server Apache**

```

<?php
function runMyFunction() {
    $output = exec('sudo python /var/www/html/ceva.py');
    //echo "$output";
    echo '<div class="location">' . $output . '</div>';

    flush();

}

if (isset($_GET['hello'])) {
    runMyFunction();
}
?>
<script>
    setInterval(function(){
        window.location.href = "index.php?hello=true";
    },1000);

</script>
<html>
    <div class = "location">
    </div>
    <table style="width: 50%; text-align: left; margin-left: auto; margin-right: auto;"
border="2" cellpadding="2" cellspacing="2">
        <tbody>
            <tr>
                <td style="text-align: center;"> <a href='index.php?
hello=true'>SHOW</a> </td>
            </tr>
        </tbody>
    </table>

```

```
</table>
</html>
```

### ceva.py

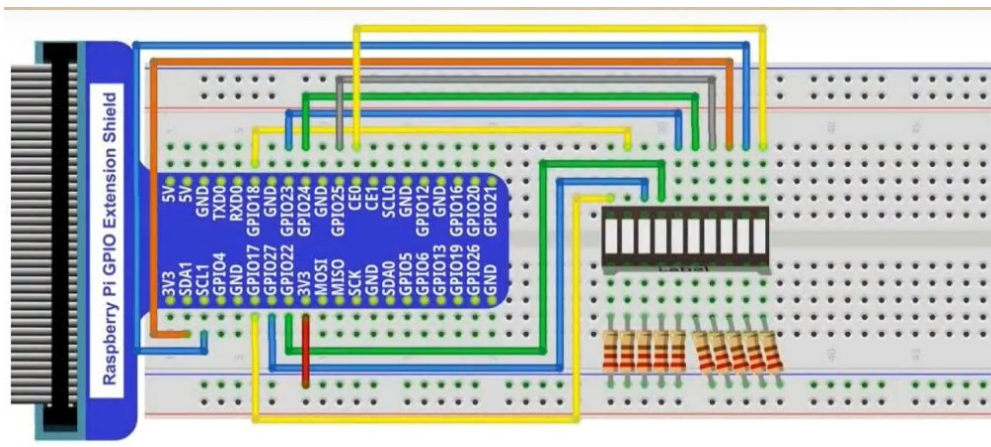
```
import serial
ser = serial.Serial('/dev/ttyUSB0')
def afiseaza():
    ser_bytes = ser.readline()
    decoded_bytes = float(ser_bytes[0:len(ser_bytes)-2].decode("utf-8"))
    dist = decoded_bytes
    print(str(dist))
    return dist
if __name__ == "__main__":
    afiseaza()
```

**Hackster.io:** <https://www.hackster.io/gems/parking-senzor-ba6af6>

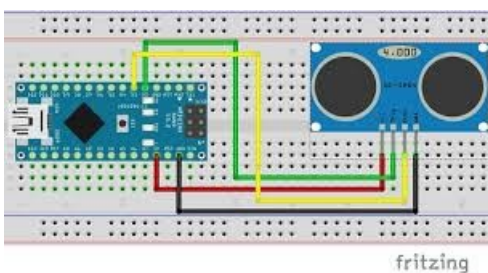
**Videoclip:** <https://youtu.be/Gx6pIWjkQYE>

**Schema:**

*Bara de leduri conectata la raspberry pi*



*Senzor conectat la arduino*



**Taskuri:**

*Simion Emilian:* Primire distanta pe seriala de la arduino, aprindere bara de leduri in functie de distanta.

*Popescu George:* Conectare senzor la arduino, trimitere distanta catre raspberry pi