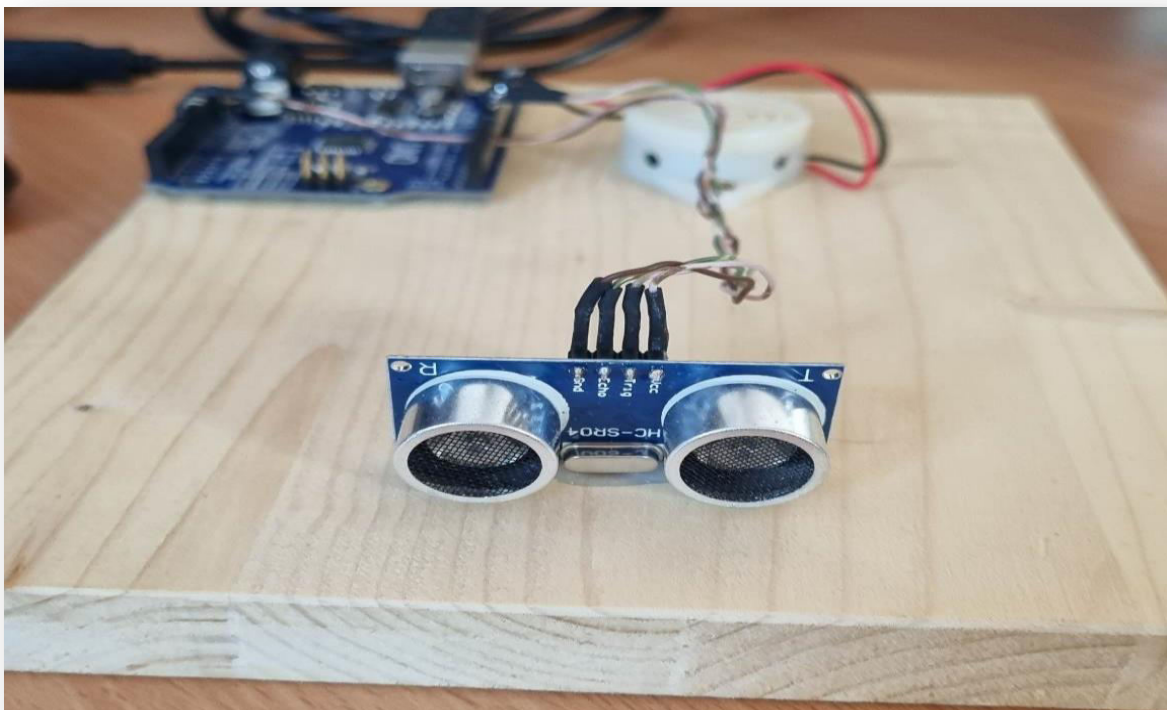
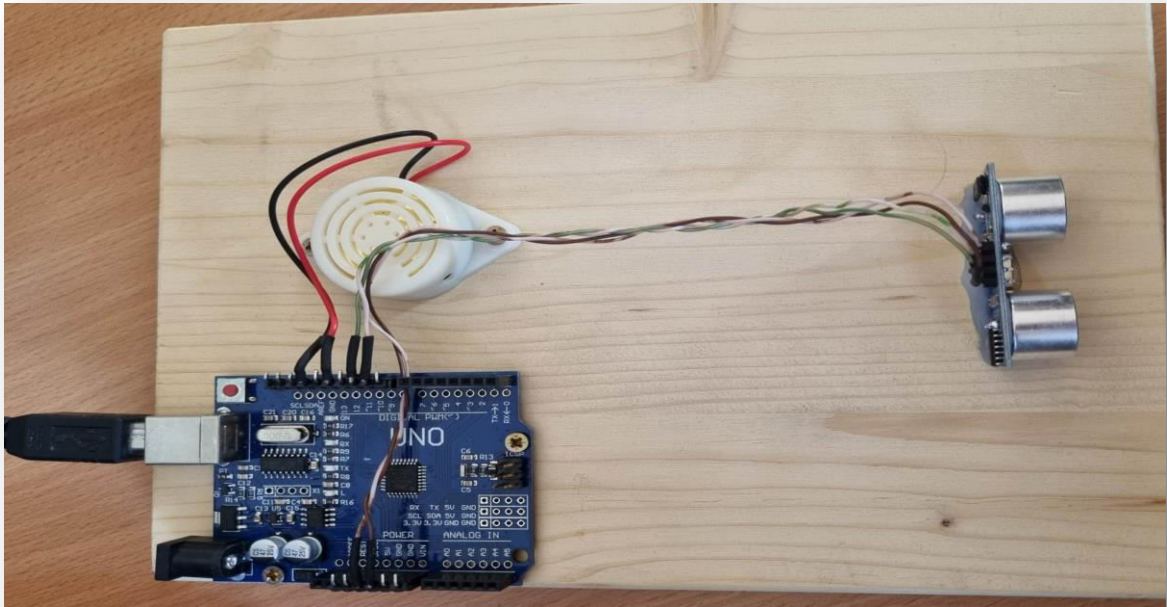

Facultatea de Automatica, Calculatoare, Inginerie Electrica si Electronica
Universitatea „Dunarea de Jos” Galati

**PROIECT ELECTRONICA DIGITALA
DETECTAREA OBSTACOLELOR SI AVERTIZAREA
ACESTORA PRINTR-UN SEMNAL SONOR**

Profesori coordonatori: Alina Pricopie Filip
Mihaela Andrei

Nume: Predi Andreea-Oana
Parteni Elena

~2023~

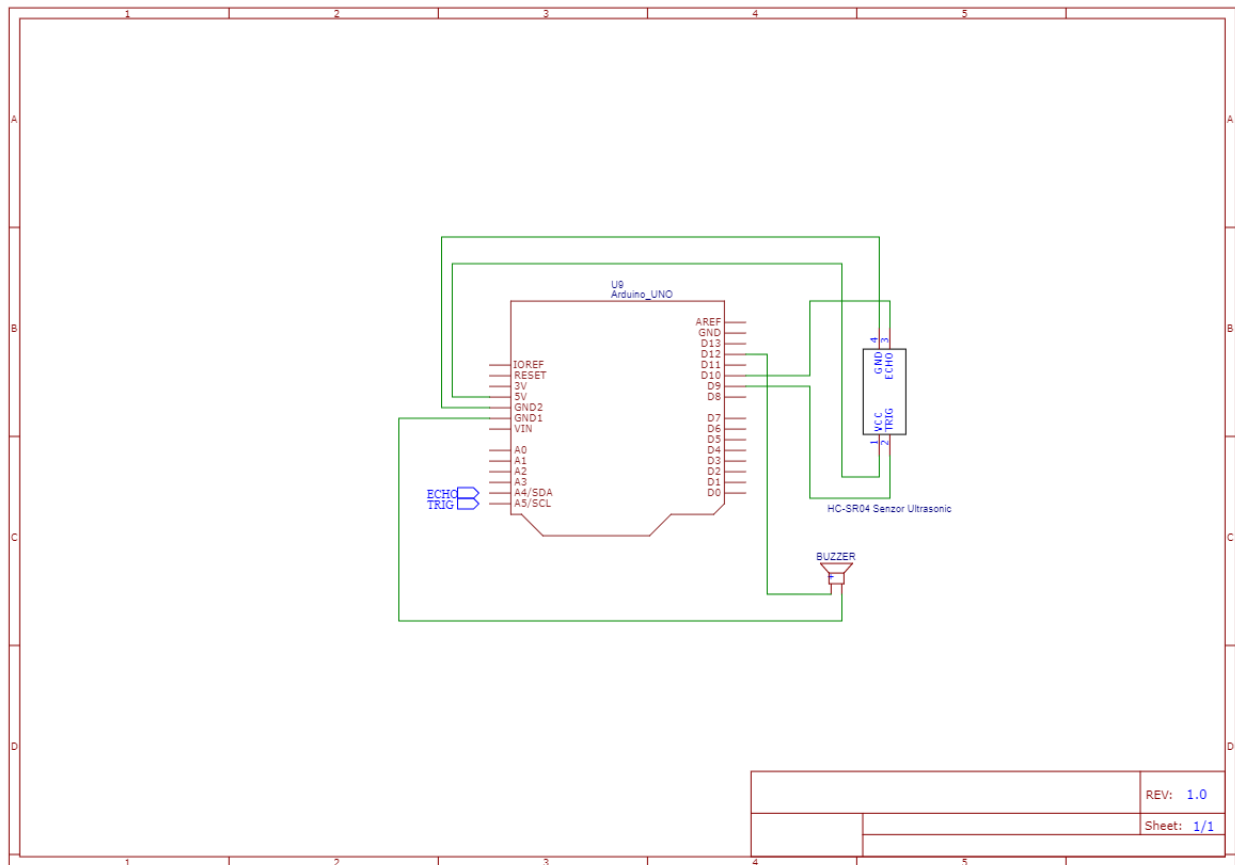


-CUPRINS-

1. Descrierea componentelor	3
2. Descrierea codului	6
3. Codul sursa	8

Componentele utilizate:

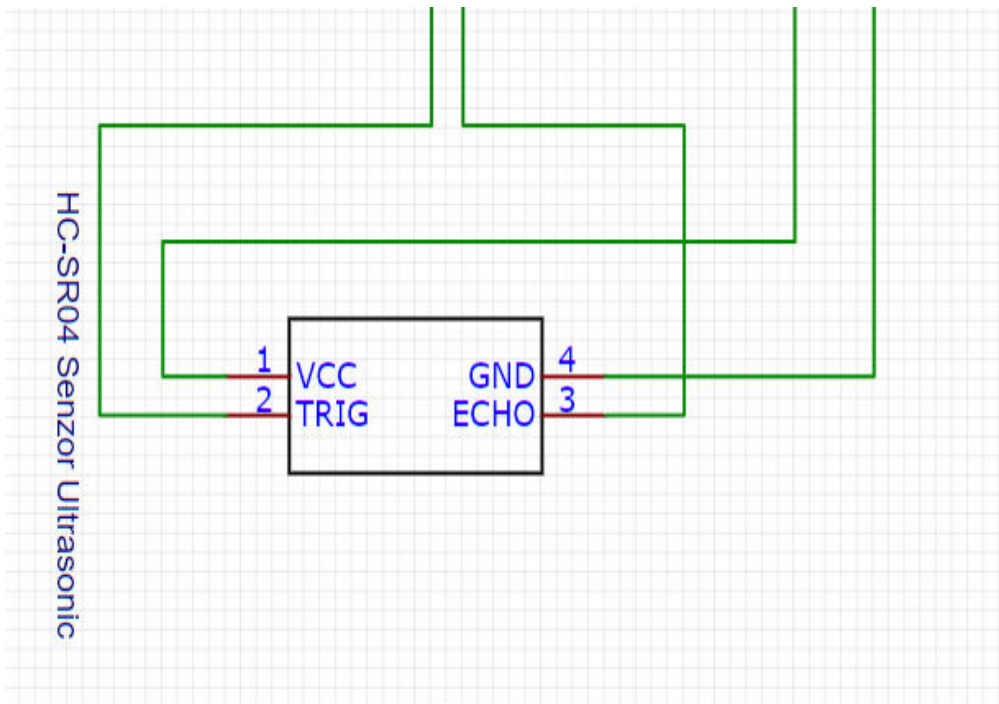
- Placa dezvoltare Arduino Uno
- Alarma Piezoelectrica
- Bareta pini la 90 de grade
- Senzor ultrasonic



Schema electrică a sistemului a fost realizată în programul Easyeda. Aici se poate vedea modul de conectare al componentelor la placa de dezvoltare. Acestea sunt alimentate de către Arduino, care este alimentat la rândul lui din cablul USB prin care se realizează și programarea.

Prima componentă este senzorul ultrasonic care are 4 pini de conectare. Primi doi sunt de alimentare, pinul VCC este conectat la 5V și GND la GND-ul microcontrolerului. Ceilalți doi pini sunt de control. Când un semnal de tensiune logică înaltă este aplicat pe pinul TRIG, senzorul trimite un ultrasunet care este reflectat de obiectul din fața lui și când senzorul îl recepționează pe pinul ECHO se trimite un semnal de tensiune logică înaltă. Știind timpul dintre semnalul de transmisie și cel de recepție se poate calcula distanța. Astfel pinul ECHO a fost conectat la pinul digital D10 și pinul TRIG la pinul digital D11.

A doua componentă este modul buzzer care acceptă o tensiune de la 3 la 24VDC, consumul de curent este foarte mic, astfel acesta a putut fi conectat direct într-un pin digital al plăci de dezvoltare. În acest mod, contactul său pozitiv a fost conectat la pinul D6 iar minusul a fost conectat la GND. Modul său de funcționare este unul simplu, cand este alimentat el vibrează în interior un mic disc piezoelectric la frecvențe înalte ce crează un sunet puternic.



```

const int TRIG_PIN    = 9;
const int ECHO_PIN    = 10;
const int BUZZER_PIN  = 12;
const int DISTANTA    = 10;

float durata_us, distanta_cm;

void setup() {
    Serial.begin (9600);
    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);
    pinMode(BUZZER_PIN, OUTPUT);
}

```

În prima parte a codului se definesc pinii de conectare care sunt salvați în variabile integer constante. Mai jos este declarată și variabila DISTANȚĂ, cand sistemul va detecta o distanță mai mică ca aceasta va porni alarma. Mai jos sunt doua variabile de tip float ca ajuta la calcularea distanței, acestea sunt salvate în acest tip de variabilă deoarece sunt numere zecimale. In final este funcția void() setup care rulează o singura data la pornire, în aceasta se declară statutul pinior de INPUT sau OUTPUT cu funcția pinMode() și se pornește comunicatia serial la viteza de 9600 cu comanda Serial.begin().

```

void loop() {

    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    durata_us = pulseIn(ECHO_PIN, HIGH);

    distanta_cm = 0.017 * durata_us;
}

```

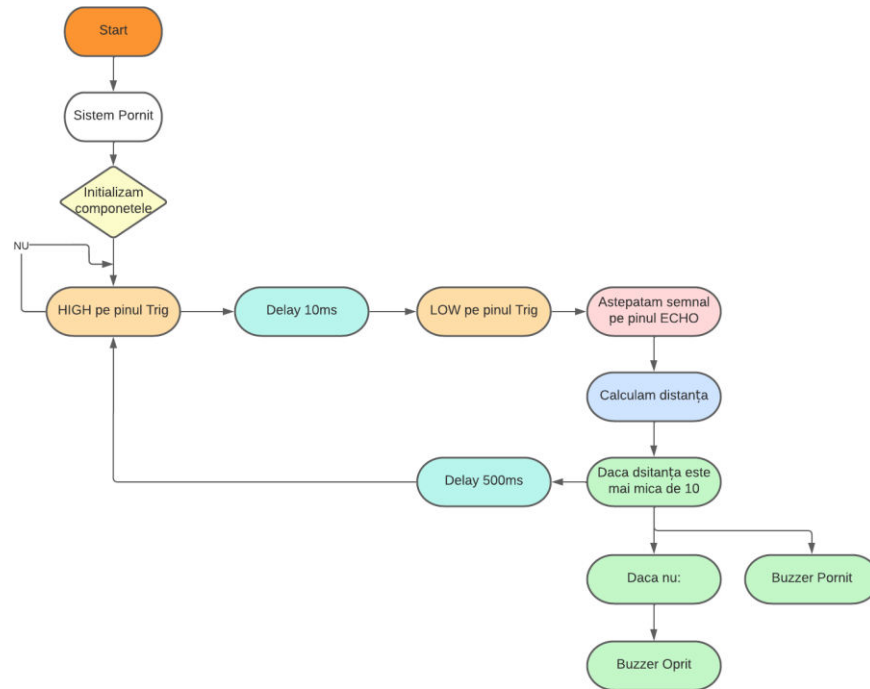
În a doua parte a codului se găsește începutul funcției void loop(), care rulează la infinit în buclă când Arduino ul este alimentată. Aici se calculează distanța după formula distanța până la obiect = ((viteza sunetului în aer)/2 * timp)/ unde viteza este de 342m/s și timpul este durata dintre transmisie și recepție a senzorului, variabila durata.

```
if(distanța_cm < DISTANTA)
    digitalWrite(BUZZER_PIN, HIGH);
else
    digitalWrite(BUZZER_PIN, LOW);

Serial.print("Distanța (Cm): ");
Serial.print(ddistanța_cm);
Serial.println(" cm");

delay(500);
}
```

În a treia parte a codului se găsește finalul funcției void loop(). Aici este conditionalul care verifică dacă distanța este mai mică de 10cm, în acest caz el pornește buzzerul cu comanda DigitalWrite(), dacă nu el stă închis. Mai jos distanța calculată este trimisă și afișată pe serial cu comanda Serial Print(). Ultimele linii de cod conțin un delay de 500ms care este activat la fiecare rulare a buclei pentru a aloca timpul necesar utilizat pentru calcul și afișare



Codul sursă:

```

const int TRIG_PIN = 9;
const int ECHO_PIN = 10;
const int BUZZER_PIN = 12;
const int DISTANTA = 10;

```

```

float durata_us, distanta_cm;

```

```

void setup() {
  Serial.begin (9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
}

```



```
void loop() {  
  
    digitalWrite(TRIG_PIN, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(TRIG_PIN, LOW);  
  
    durata_us = pulseIn(ECHO_PIN, HIGH);  
  
    distanza_cm = 0.017 * durata_us;  
  
    if(distanza_cm < DISTANTA)  
        digitalWrite(BUZZER_PIN, HIGH);  
    else  
        digitalWrite(BUZZER_PIN, LOW);  
  
    Serial.print("Distanza (Cm): ");  
    Serial.print(distanza_cm);  
    Serial.println(" cm");  
    delay(500); }
```