

Sistem Radar cu Interfata Grafica

Introducere

Proiectul propune implementarea unui sistem radar adaptat pentru utilizare la scară mică, inspirat de conceptele de monitorizare a distanțelor. Un senzor de distanță ultrasonic, cum ar fi senzorul HC-SR04, montat pe un servomotor, este utilizat pentru a măsura distanțele înconjurătoare. Obiectele detectate sunt vizualizate pe un calculator printr-o interfață grafică simplă. Utilizatorul este notificat atât vizual, cât și auditiv, de către un buzzer. Această soluție oferă o metodă completă și eficientă pentru monitorizarea și detectarea obiectelor din proximitate.

Scopul fundamental al acestui proiect este de a furniza un cadru inițial pentru realizarea unor proiecte mai complexe, demonstrând în același timp cât de ușor poate fi implementat un sistem de achiziție de date folosind Arduino și senzori compatibili, precum senzorul HC-SR04. Datele sunt prelucrate local de placa Arduino, apoi transmise către un desktop PC pentru afișarea lor într-un format accesibil și intuitiv.

Proiectul s-a născut din dorința de a explora și de a înțelege potențialul sistemelor de monitorizare și detecție a obiectelor într-un context accesibil și practic. Adaptând conceptele tehnologice pentru utilizare la scară mică, acest proiect oferă oportunitatea de a experimenta și de a învăța despre interconectarea componentelor hardware și software.

Descriere generala

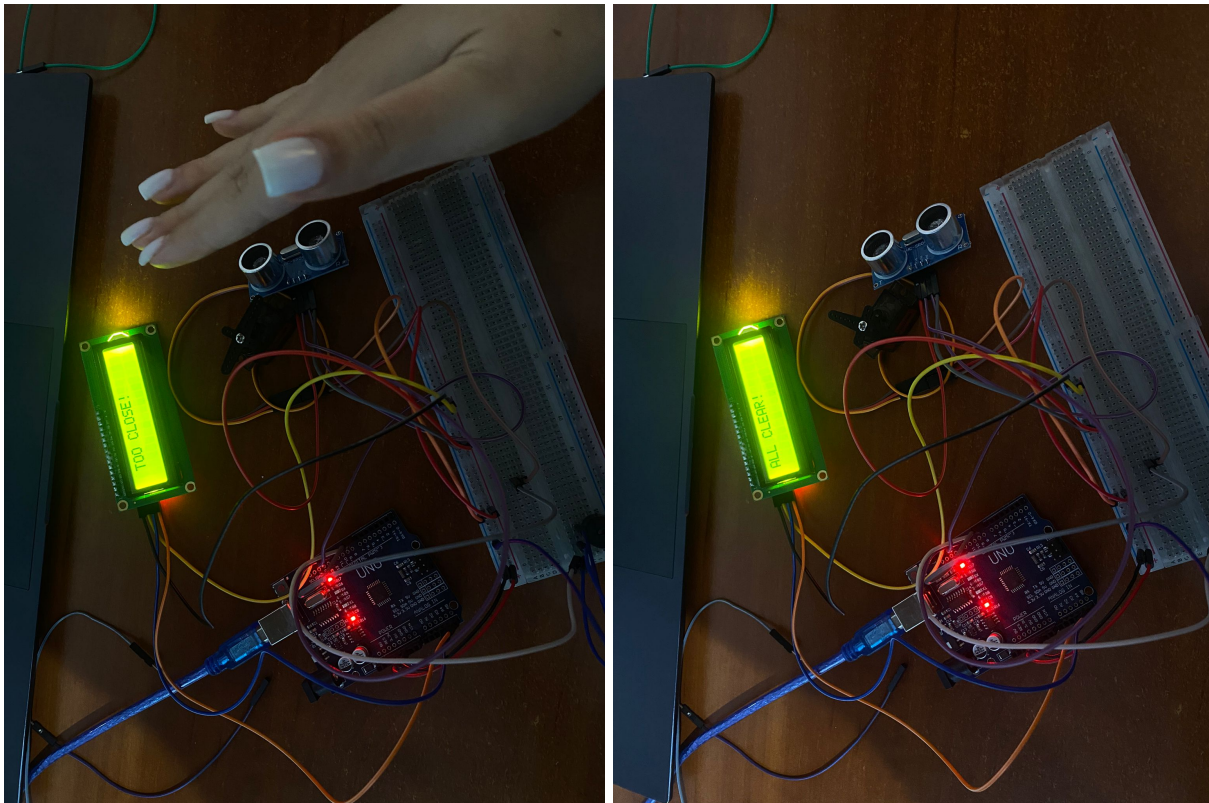
Acest proiect implementează un sistem radar utilizând Arduino UNO, senzorul de distanță HC-SR04 montat pe un suport special pe un servomotor. Senzorul HC-SR04 măsoară distanțele înconjurătoare, iar servomotorul rotește radarul intermitent pentru a scana zona de interes. Atunci când un obiect este detectat în raza senzorului, un buzzer emite un sunet de atenționare, iar un LED RGB se aprinde: roșu dacă distanța este sub o anumită valoare limită (THRESHOLD), galben dacă distanța este egală cu această valoare limită și verde în restul cazurilor. Informațiile despre obiectele detectate sunt afișate pe un laptop printr-o interfață grafică simplă și intuitivă. De asemenea, un mesaj intuitiv este afișat pe un display LCD în cazul în care este detectat un obiect în raza de acoperire, astfel:

- *TOO CLOSE!* - în cazul în care se află un obiect la o distanță mai mică decât limita impusă (25 cm);
- *A LITTLE CLOSE!* - în cazul în care se află un obiect la o distanță intermediară ($25 < \text{distance} < 50$).
- *ALL CLEAR!* - în cazul în care se află un obiect la o distanță mai mare decât limita impusă (50 cm).

Schema bloc



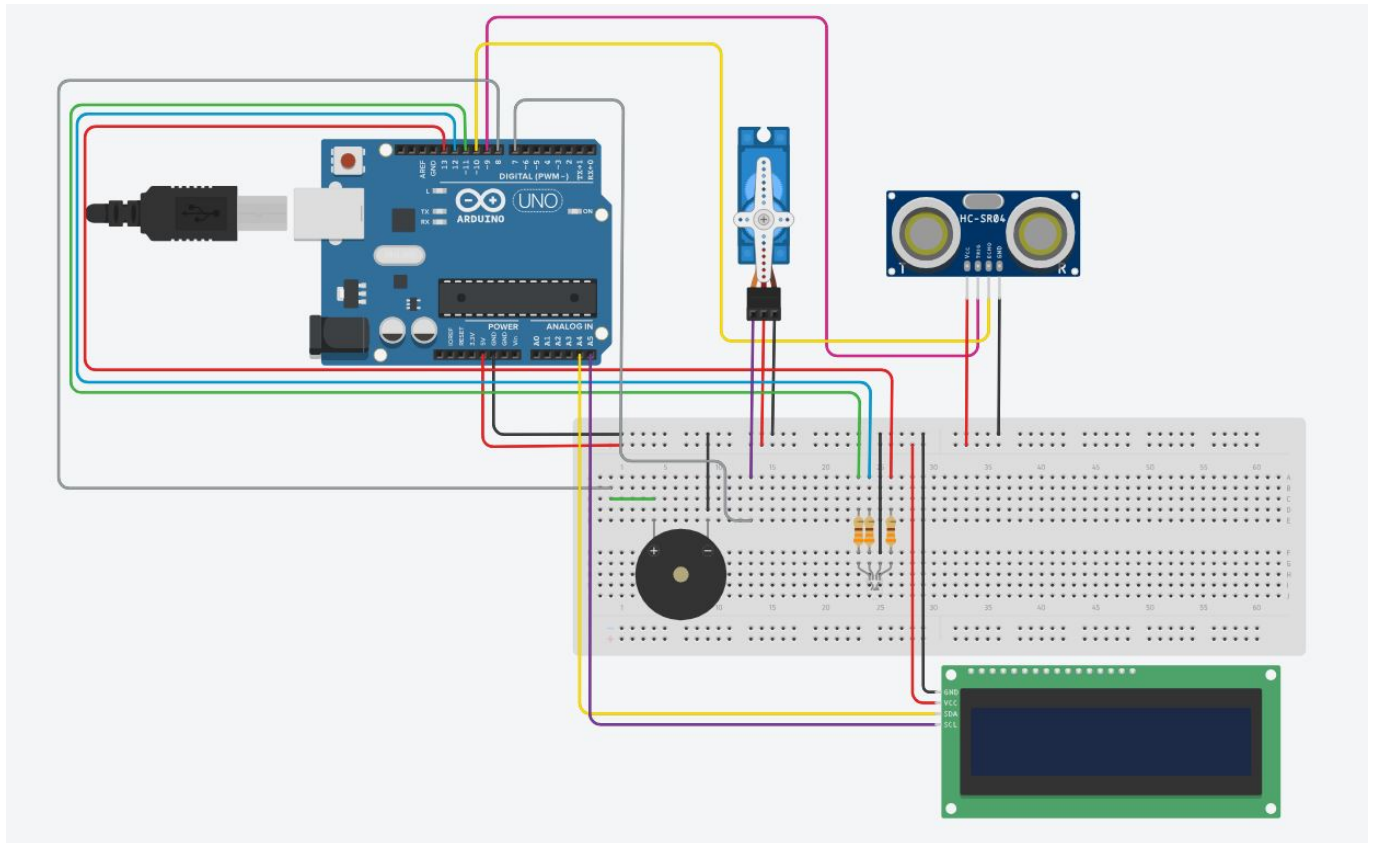
Hardware Design



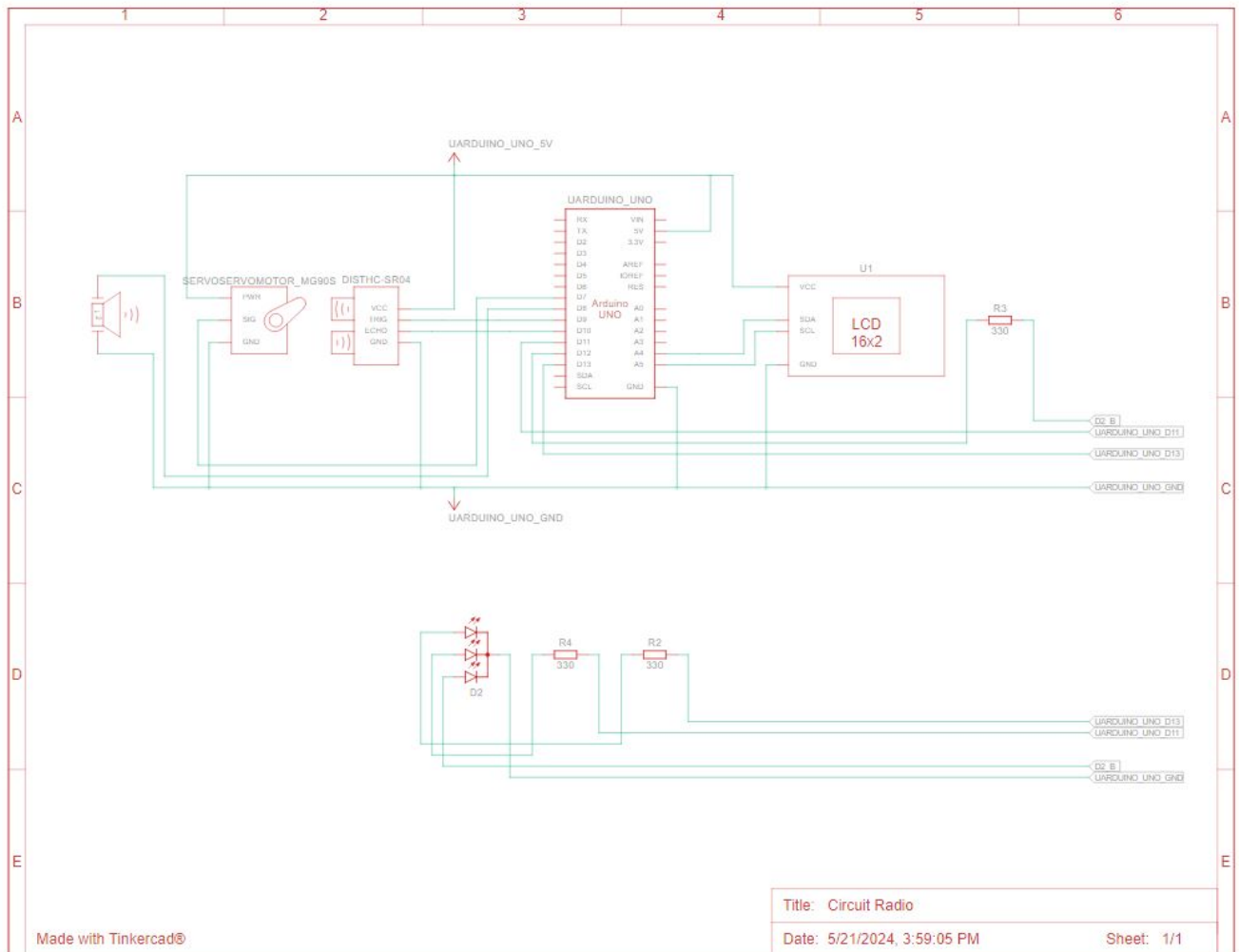
Componente Hardware utilizate:

- Arduino UNO
- Breadboard
- Servomotor MG90S
- Senzor de proximitate HC-SR04
- Buzzer
- Led RGB
- LCD

Schema Arduino Tinkercad:



Schema electrica:



Schema de Conectare

1. Arduino UNO:

- Toate conexiunile componentelor sunt realizate cu pinii digitali și analogici ai plăcii Arduino UNO.

2. Breadboard:

- Distribuie +5V și GND pentru componentele care necesită aceste conexiuni.
- Utilizat pentru legarea servomotorului și a buzzerului la pinii respectivi.

3. Servomotor MG90S:

- Semnal: Pin 8
- Vcc: +5V prin breadboard
- Ground: GND prin breadboard

4. Senzor de proximitate HC-SR04:

- Echo: Pin 10
- Trig: Pin 9

- Vcc: +5V direct la Arduino
- Ground: GND direct la Arduino

5. Buzzer:

- Semnal: Pin 8 (prin breadboard)
- Ground: GND prin breadboard

6. LED RGB:

- Roșu: Pin 13 prin rezistență de 330Ω
- Verde: Pin 11 prin rezistență de 330Ω
- Albastru: Pin 12 prin rezistență de 330Ω
- Ground: GND (prin breadboard)

7. Display LCD:

- SCL: Pin A4
- SDA: Pin A5
- Vcc: +5V prin breadboard
- Ground: GND prin breadboard

Software Design

Medii de dezvoltare:

- Cod: ArduinoIDE
- Schema bloc: drawIO
- Schema electrică: TinkerCad
- Design circuit: TinkerCad
- Biblioteci folosite:
 - LiquidCrystal_I2C.h: pentru a controla ecranul LCD cu interfață I2C.
 - Servo.h: pentru a controla servomotorul.

1. Codul Arduino: În cadrul codului Arduino, servomotorul rotește senzorul ultrasonic și măsoară distanța față de obiecte. Pe baza distanței măsurate, LED-ul își schimbă culoarea, buzzer-ul sună și mesajul de pe ecranul LCD se actualizează. De asemenea, datele despre unghi și distanță sunt trimise către interfața serială pentru a fi prelucrate ulterior de aplicația GUI.

2. Funcționare: În cadrul funcției loop, la fiecare pas de rotație al servomotorului, se apelează funcția distanceMeasurement(), unde se primesc datele de la senzorul de distanță ultrasonic. Dacă distanța măsurată este mai mică decât limita impusă, atunci:

- Culoarea default a LED-ului (verde) se schimbă în roșu.
- Buzzer-ul emite un sunet.
- Mesajul default de pe ecranul LCD ("ALL CLEAR!") se schimbă în funcție de distanță ("TOO CLOSE!" sau "A LITTLE CLOSE!").
- Pe interfața serială sunt trimise datele despre unghi și distanță pentru fiecare iterație a

servomotorului în care a fost detectat un obiect.

Inițializarea și configurarea piniilor:

Funcția `initializePins()` configurează pinii pentru trig, echo, buzzer și LED-ul RGB ca OUTPUT.

Scanarea și măsurarea distanței:

Funcția `scanAndMeasure()` rotește servomotorul de la 0 la 170 de grade și înapoi, măsurând distanța la fiecare pas. Funcția `distanceMeasurement()` măsoară distanța folosind senzorul ultrasonic, trimite datele prin serial și actualizează ecranul LCD și culoarea LED-ului.

Afișarea mesajului și setarea culorii LED-ului:

Funcția `displayMessageAndColor()` afișează un mesaj pe LCD și setează culoarea LED-ului în funcție de distanța măsurată. Funcția `setColor()` setează intensitatea LED-urilor RGB pentru a afișa culoarea corespunzătoare (roșu, portocaliu sau verde).

3. Codul Java folosind framework-ul de Processing:

Serial Communication:

- Se inițializează comunicarea serială pentru a primi date de la un dispozitiv radar, conectat la portul COM5 și setat la o viteză de transfer de 9600 baud.

Desenarea Elementelor Vizuale:

- `drawRadar()`: Desenează arcele concentrice și liniile radiale pentru a reprezenta zona de scanare a radarului.
- `drawObject()`: Detectează și afișează obiectele detectate pe radar în funcție de distanță și unghi.
- `drawLine()`: Desenează linia rotativă a radarului, indicând direcția de scanare.
- `drawText()`: Afișează informații despre distanță, unghi și starea de siguranță a mediului înconjurător.

Manipularea Datelor de la Radar:

- `serialEvent()`: Este activată atunci când sunt primite date de la radar prin comunicarea serială. Aceasta interpretează datele primite și extrage unghiul și distanța obiectelor detectate.

Desenarea Componentelor Vizuale:

- `drawArcs()`: Desenează arcele concentrice care reprezintă diferitele intervale de distanță ale radarului.
- `drawAngleLines()`: Desenează liniile radiale care reprezintă diferitele unghiuri de scanare ale radarului.
- `drawAngleLabels()`: Desenează marcasele de unghiuri pe radar pentru a indica direcția de scanare.

Afișarea Informațiilor:

- `setObjectColor()`: Stabilește culoarea obiectelor detectate în funcție de distanță.
- `drawAngleLabel()`: Desenează etichetele de unghi pentru a indica direcția de scanare a radarului.

Rezultate Obținute

Video cu sistemul radar: <https://youtube.com/shorts/3qU8o9yFfes?si=KjTPv5YFtWFnj9V1>

Concluzii

A fost o provocare sa implementez primul meu proiect de acest gen, dar si o introducere utila in acest domeniu.

Proiectul este functional si pregatit sa serveasca la detectia obiectelor pe o raza minimala.

Download

Codul sursa al proiectului:

[georgescu_andreea.daniela.zip](#)

Jurnal

- 25 aprilie: Alegere finala proiect.
- 30 aprilie: Am comandat piesele.
- 3 mai: Sosire comanda.
- 7 mai: Comanda suplimentară pentru piese.
- 10 mai: Sosire comanda suplimentară.
- 16 mai: Milestone Hardware.
- 23 mai: Milestone Software

Bibliografie/Resurse

In realizarea proiectului m-am folosit de informatii preluate din urmatoarele surse:

Resurse Hardware:

- <https://www.geeksforgeeks.org/how-to-interface-i2c-lcd-display-with-arduino/>
- <https://www.youtube.com/watch?v=n-gj00GTsNg>
- <https://www.ardumotive.com/how-to-use-a-buzzer-en.html>

Resurse Software:

- <https://www.youtube.com/watch?v=w5-LD80GJNs>
- <https://chat.openai.com/>

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2024/amocanu/andreea.georgescu02>



Last update: **2024/05/26 22:47**