

Sistem radar de atentionare (military grade)



Author: [Cristian-Tanase Paris](#)

Introducere

Acest proiect implementeaza un sistem radar similar radarelor militare de pe nave. Sistemul se bazeaza pe un senzor ultrasonic montat pe un servomotor pentru a spori raza de actiune. Obiectele din raza senzorului vor fi afisate pe un calculator intr-o interfata grafica similara radarelor navale si va exista un sistem de notificare vizual/acustic.

Scopul acestui proiect este de a oferi un punct de start pentru alte proiecte mai ample si a dovedi usurinta implementarii unui sistem complet de achizitii de date folosind Arduino si diversii senzori compatibili. Astfel, procesarea datelor este realizata de placuta si acestea sunt trimise apoi unui desktop PC pentru afisarea informatiilor relevante.

Ideea de baza de la care am pornit a fost un documentar legat de navele de razboi si sistemele de detectie radar/sonar utilizare pentru gasirea inamicilor. Astfel, am considerat ca acest model poate fi implementat relativ usor la o scara mica prin intermediul Arduino.

Consider acest proiect util deoarece reprezinta o combinatie armonioasa atat a partii hardware, cat si software in realizarea unui sistem care poate avea aplicabilitate in cadrul unor sisteme mai complexe (ex. alarma de proximitate, iluminare inteligenta, etc.)

Descriere generală

Distanța limită pentru avertizarea acustică/vizuală este presetată by default la 10 cm. Comunicatia dintre Arduino si PC se va realiza prin intermediul interfetei seriale. Senzorul(aflat într-un suport prins de servomotor) va trimite datele de distanță la Arduino, care le va prelucra si va verifica dacă există vreun obiect care a intrat în raza minimă de avertizare.

Dacă un obiect intra în raza limită de avertizare, un led roșu se va aprinde intermitent, buzzer-ul se va activa si pe un ecran lcd se va afișa un mesaj sugestiv. Totodata, pe interfata radar de pe PC, respectivul obiect va fi evidențiat într-o altă culoare.

În materie de soft, interfata va fi realizată în framework-ul educational de vizualizare scris în Java, Processing.

Schema bloc:



Hardware Design

Lista de piese:

- Arduino UNO Rev.3
- Ecran LCD
- Buzzer
- Servomotor
- Rezistente
- Diverse tipuri de LED-uri
- Fire
- Senzor de distanta ultrasonic HC-SR04
- Holder pentru senzor
- Breadboard

Schema electrica:



Design-ul circuitului:



Software Design

* Medii de dezvoltare:

- Cod: ArduinoIDE
- Schema bloc: drawIO
- Schema electrica: EAGLE CAD
- Design circuit: TinkerCad

* Biblioteci folosite:

- LiquidCrystal_I2C.h
- Servo.h

* Codul Arduino:

In cadrul functiei loop, la fiecare pas de rotatie al servomotorului, se va apela functia distanceCheck(), in cadrul careia se vor primi datele de la senzorul de distanta ultrasonic. Daca distanta calculata pe baza datelor este mai mica decat limita impusa, atunci culoarea default a led-ului (care este verde si se reseteaza la fiecare iteratie) va deveni rosie, buzzer-ul va tiui, iar textul default de pe ecranul LCD ("NO OBJECTS ENC") se va schimba ("OBJECT IN RANGE"). Totodata, pe interfata seriala sunt trimise datele legate de unghi si distanta pentru fiecare iteratie a servo-ului in care a fost detectat un obiect.

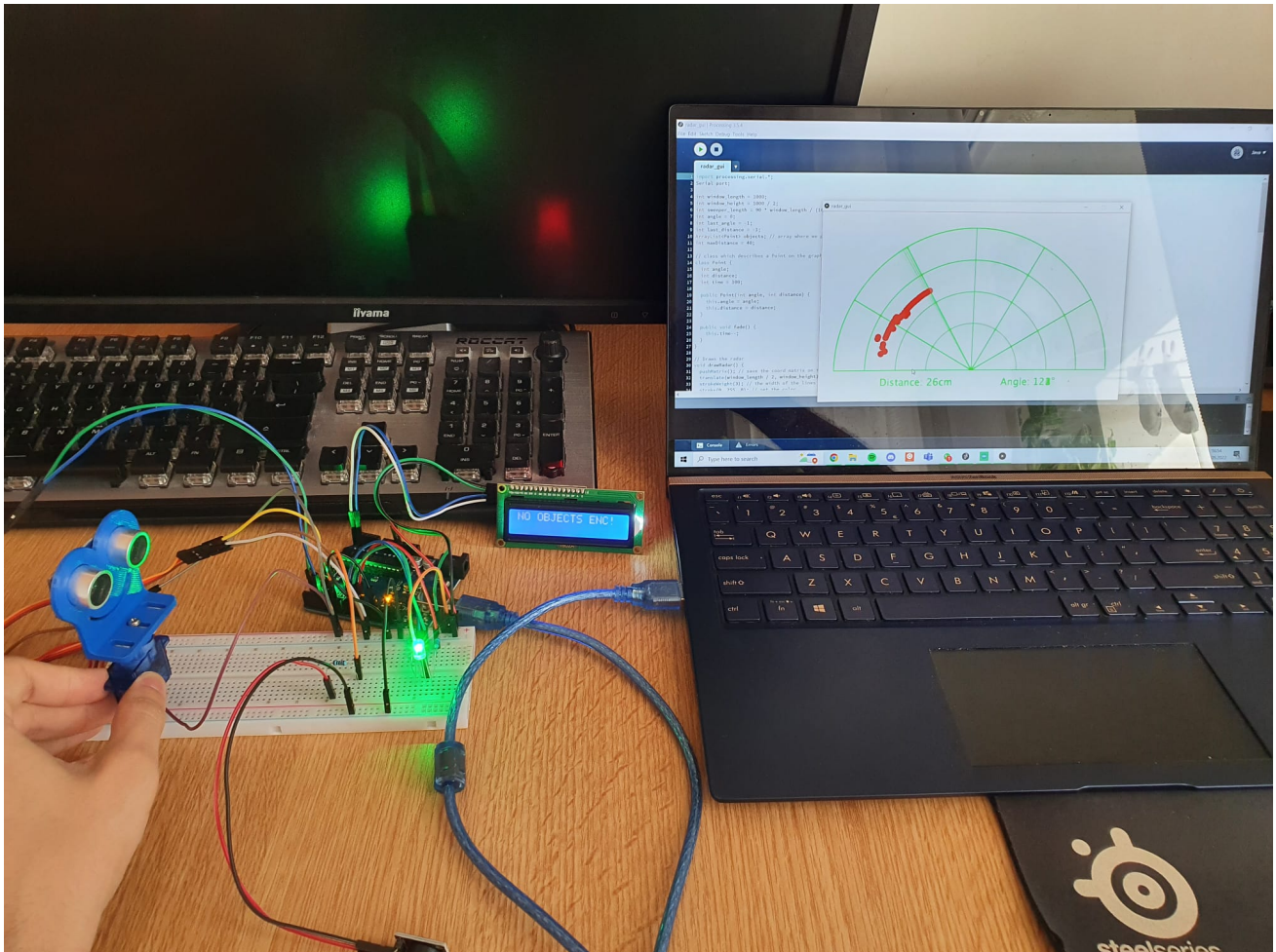
Este de notat faptul ca am reusit in cadrul codului de pe placuta sa nu introduc niciun delay, in afara de cele aferente senzorului de distanta. Acest lucru duce la o mai buna utilizare a microprocesorului si la un sweep al radarului mai putin sacadat. Totodata, codul poate fi astfel extins cu alte functionalitati in paralel.

* Codul Java folosind framework-ul de Processing:

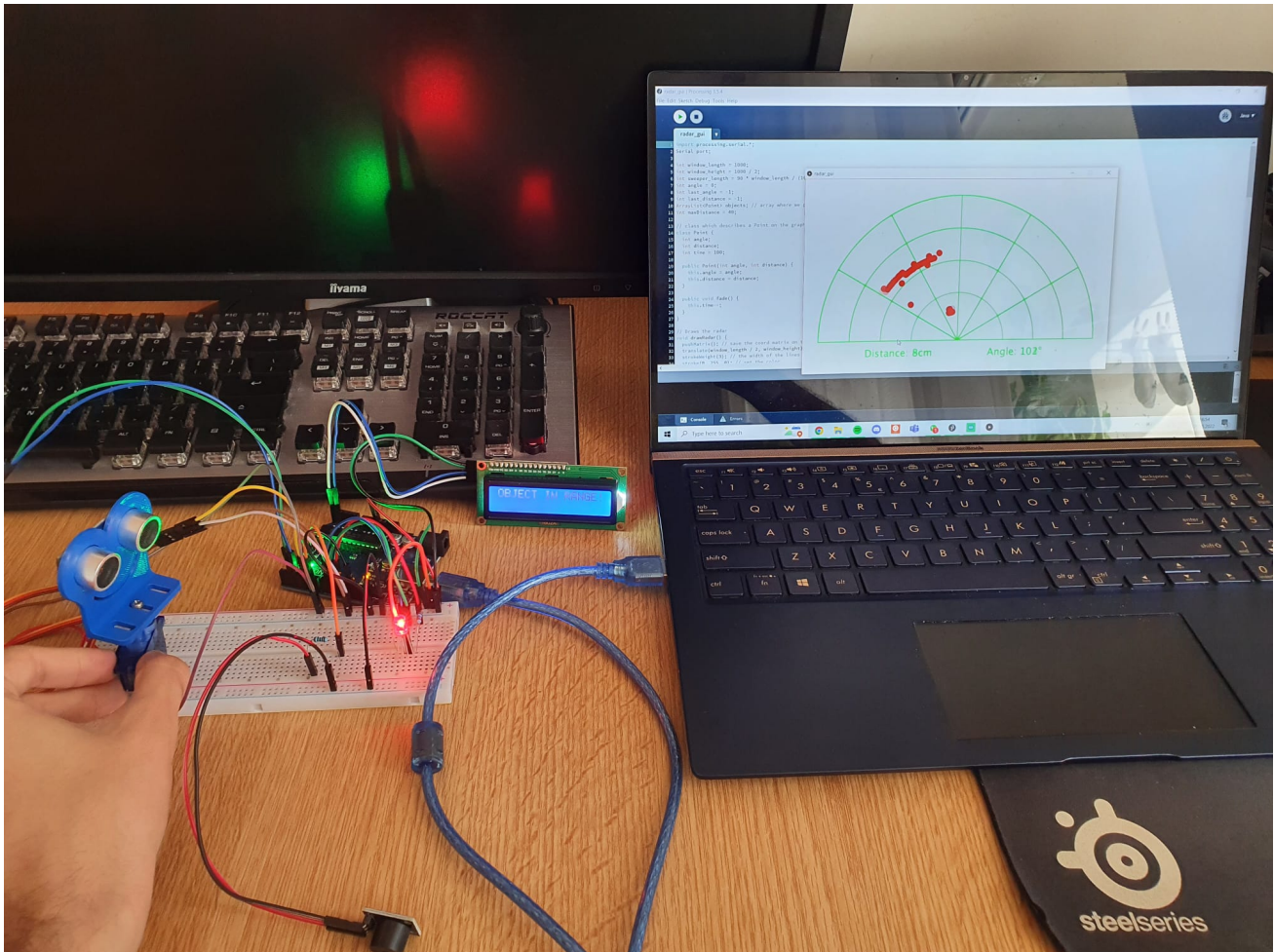
Aplicatia GUI primeste prin intermediul portului serial datele de pe placuta intr-un format prestabilit. Datele sunt parsate si sunt impartite in partea de distanta si de unghi. Daca distanta primita este mai mica decat limita impusa in cod, atunci se va crea un nou obiect de tip punct. Toate aceste obiecte vor fi apoi desenate in cadrul functiei drawObjects, impreuna cu unghiul si distanta la care s-au facut detectia. Restul radarului se va redesena la fiecare iteratie.

Rezultate Obținute

In aceasta poza putem observa comportamentul atunci cand niciun obiect nu e in raza de detectie:



lar in aceasta poza atunci cand un obiect este detectat:



Totodata, aveti [aici](#) un video in care se poate observa mai clar modul de functionare.

Mici observatii

Sistemul functioneaza exact dupa cum l-am gandit, singura problema este ca servo-ul se misca mult mai incet decat planuisem initial si decat il testasem. Acest lucru se datoreaza faptului ca Arduino-ul nu poate furniza destul curent pentru a tine servo-ul, buzzer-ul, led-ul si display la capacitate maxima. Totodata, uneori display-ul are un flickering la miscarea sacadata a servo-ului, datorat celor mentionate anterior.

De asemenea, din cauza lipsei de curent, uneori se intampla ca Arduino-ul sa nu furnizeze pe seriala si datele legate de unghi GUI-ului (trimite null string). Am tratat acest impediment in cod-ul GUI-ului, ignorand acele date incomplete.

Concluzii

Acest proiect a fost foarte enjoyable deoarece a imbinat atat o parte hardware, cat si software. Am apreciat faptul ca am putut sa aplic ce am invatat in cadrul laboratoarelor si ca, avand putina rabdare atat pe partea de debugging cat si pe partea de cautat pe Internet, poti sa creezi aproape orice.

Download

Arhiva cu resursele folosite se poate descarca de [aici](#).

Bibliografie/Resurse

- [Proiect similar](#)
- [ATMega-328P Datasheet](#)
- [Processing IDE](#)
- [Arduino IDE](#)
- [Configurare buzzer](#)
- [Configurare servo](#)
- [Configurare LCD I2C](#)
- [Configurare senzor ultrasonic](#)
- [Biblioteca pentru servo](#)
- [Biblioteca pentru display](#)
- [Folosirea functiei millis\(\)](#)
- [Servo sweep fara delay-uri in cod](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/rstanescu/cristian.paris>



Last update: **2022/06/01 18:05**