

APLICAȚIE RADAR CU UN SENZOR ULTRASONIC, UN SERVOMOTOR ȘI O APLICAȚIE GRAFICĂ

Realizat de Haiș Sebastian și Pătruț Deiana

1. Introducere

Proiectul nostru presupune construirea unei aplicații radar care măsoară o distanță calculată permanent de un senzor ultrasonic. Senzorul se rotește în jurul axei sale de rotație cu ajutorul unui servomotor pentru a avea o vizibilitate în jur de 160°.

Am ales această temă pentru a putea ilustra îmbinarea dintre hardware și software pentru o problemă comună și utilă în practică. Pentru că putem calcula distanțele față de obiecte foarte rapid prin intermediul acestui senzor, putem spune că prototipul nostru este „un radar în timp real” care poate fi utilizat în aplicații complexe, inclusiv în cartografie. Îmbunătățirile aduse constau în faptul că radarul nostru este unul interactiv, deoarece distanța față de fiecare obiect va fi procesată și utilizată pentru a realiza o aplicație grafică. De asemenea, prezența servomotorului permite o vizibilitate mult mai bună (raportată la componentele hardware folosite).

2. State of the art

Două aplicații asemănătoare care stau la baza inspirației pentru proiectul nostru sunt Radarul marin și Radarul de supraveghere folosit în aeroporturi:

- **RADARUL MARIN**- Radarele marine sunt radare cu bandă X sau bandă S de pe nave, utilizate pentru detectarea altor nave și obstacole terestre cu scopul de a asigura unghiul și distanța pentru evitarea coliziunilor și navigarea pe mare. Sunt instrumente electronice de navigație care utilizează o antenă rotativă pentru a transmite un fascicul îngust de microunde în jurul suprafeței apei care înconjoară nava până la orizont, detectând ținte prin microundele reflectate de la acestea și generând o imagine a împrejurimilor navei pe un ecran de afișare [1].
- **RADARUL DE SUPRAVEGHERE AEROPORT**- Un ASR este un sistem radar utilizat de aeroporturi pentru a detecta și afișa prezența și poziția aeronavelor în zona terminalului, spațiul aerian din jurul aeroporturilor. Pe aeroporturile mari se controlează de obicei traficul pe o rază de 96 de mile de aeroport, sub o altitudine de 25.000 de picioare. Sistemele sofisticate de pe aeroporturile mari constau în două sisteme radar diferite: radarul de supraveghere primar și cel secundar. Radarul primar reprezintă o antenă parabolică rotativă mare care transmite un fascicul vertical de microunde în jurul spațiului aerian care înconjoară aeroportul. Aceasta detectează poziția și raza de acțiune a microundelor reflectate înapoi la antenă de la suprafața aeronavei. Radarul secundar de supraveghere constă dintr-o a doua antenă rotativă ce transmite un semnal radio înapoi care conține identificarea aeronavei, altitudinea barometrică și un cod de stare de urgență care este afișat pe ecran împreună cu poziția aeronavei [2].

Radarul propus de proiectul nostru se aseamănă cu cele două radare din practică prezentate mai sus prin principiul de funcționare: măsurarea unei distanțe. Radarul marin și cel pentru aeroporturi se folosesc de o antenă rotativă pentru a măsura, în schimb radarul

nostru utilizează un senzor ultrasonic aflat pe un servomotor, având astfel o vizibilitate mai redusă. Distanțele ce pot fi măsurate diferă datorită echipamentelor folosite. Toate cele trei radare utilizează informația obținută pentru afișarea pe un ecran, dar în moduri diferite.

3. Design și implementare

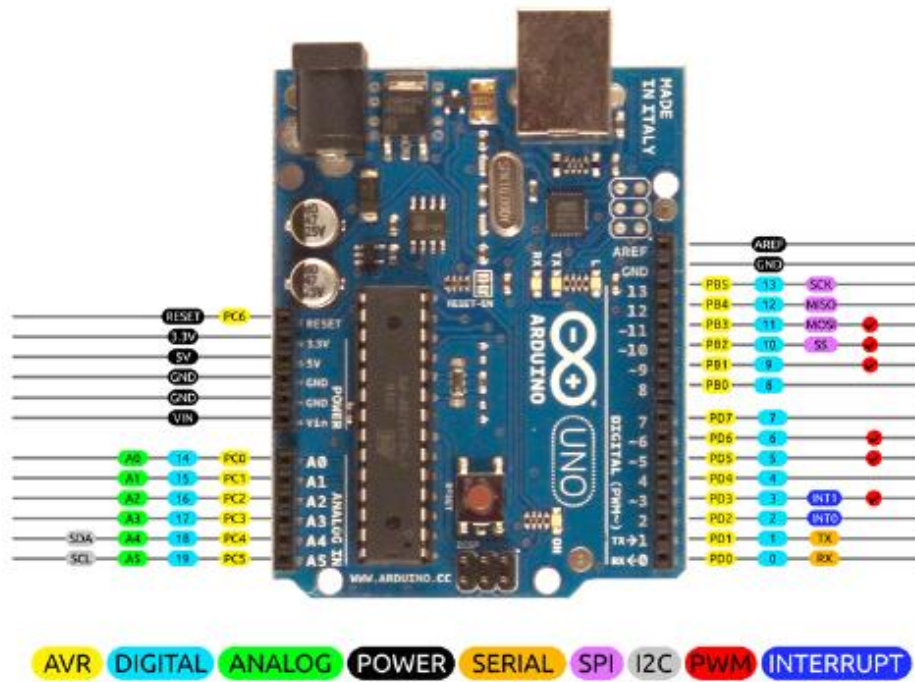
3.1. Componente

a. PLACA ARDUINO UNO

Aplicația noastră se folosește de un servomotor pentru un câmp vizual cât mai mare, acesta fiind controlat de o placă Arduino. Legătura dintre semnalele de pe placă și semnalele de la dispozitivele externe sunt făcute de un shield-buddy.

Arduino este o companie de hardware și software open source, comunitate de proiecte și utilizatori care proiectează și produce microcontrolere cu o singură placă și kituri de microcontrolere pentru construirea de dispozitive digitale. Arduino Uno este unul dintre numeroasele plăci oficiale produse de compania italiană Smart Projects. Are 14 pini digitali de intrare / ieșire (din care 6 pot fi folosiți ca ieșiri PWM), 6 intrări analogice, un rezonator ceramic de 16 MHz, o conexiune USB, o mufă de alimentare, un antet ICSP și un buton de resetare. Conține tot ce trebuie pentru a suporta microcontrolerul; pur și simplu se conectează la un computer cu un cablu USB sau se alimentează cu un AC-to-DC adaptor sau baterie pentru a începe [3].

Pin Category	Pin Name
Power	V_{in} , 3.3V, 5V, GND
Reset	Rst
Analog Pins	A0-A5
Input/Output Pins	Digital Pins 0-13
Serial	0(R_x), 1(T_x)
External Interrupts	2,3
PWM	3,5,6,9,11
SPI	10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)
Inbuilt LED	13



b. MODUL CU ULTRASUNETE HC - SR04

Modulul cu ultrasunete HC-SR04 este folosit pentru a oferi funcția de măsurare fără contact, intervalul de precizie putând ajunge la 3 mm. Modulele includ emițătoare cu ultrasunete, receptor și circuit de control.

Principiul de bază al funcționării este:

- (1) Utilizarea declanșatorului IO pentru un semnal de nivel înalt de cel puțin 10 μ s
- (2) Modulul trimite automat opt 40 kHz și detectează dacă există un semnal de impuls înapoi
- (3) Dacă semnalul înapoi există, la nivel ridicat, timpul cu o durată mare de ieșire IO este timpul de la trimiterea ultrasunetelor până la întoarcere [4].

Conectarea firelor este după cum urmează:

- Alimentare 5V
- Trigger Pulse Input
- Ieșire Echo Pulse
- Masă 0V

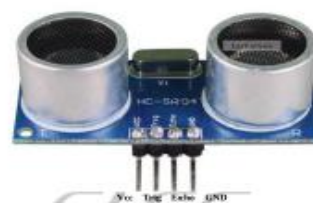
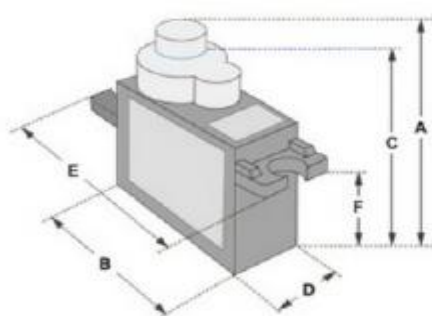


Fig. 3: Modulul cu ultrasunete HC-SR04

c. SERVOMOTORUL SG90

Un servomotor este un dispozitiv care se poate roti cu aproximativ 180 de grade (90 în fiecare direcție) și funcționează la fel ca tipurile standard dar mai mici. Controlul unui servo se poate face folosind cod hardware sau cod dintr-o bibliotecă. Deoarece este simplu de înțeles, vom folosi biblioteca Arduino „Servo.h”. Servomotorul are 3 fire: putere, masă și semnal. Firul de alimentare este de obicei roșu și ar trebui să fie conectat la pinul de 5V de pe placa Arduino. Firul de masă este de obicei negru sau maro și ar trebui să fie conectat la un pin de masă de pe placa Arduino. Pinul de semnal este de obicei galben, portocaliu sau alb și trebuie conectat la un pin digital de pe placa Arduino [5].



Dimensions & Specifications	
A (mm) :	32
B (mm) :	23
C (mm) :	28.5
D (mm) :	12
E (mm) :	32
F (mm) :	19.5
Speed (sec) :	0.1
Torque (kg-cm) :	2.5
Weight (g) :	14.7
Voltage :	4.8 - 6

Fig. 4: Servomotorul SG90

d. PLACA DE EXPANSIUNE SENSOR SHIELD V5.0

Această placă-scut scoate pinii standard I/O ai lui Arduino la anteturi, împreună cu pinii de masă dedicați și pinii de putere per I/O pentru a facilita conectarea senzorilor la alte dispozitive. Scutul are, de asemenea, o serie de conectori dedicați care sunt definiți în scopurile specifice, așa cum sunt prezentate în figura 5, , dar pot fi considerați și puncte de conexiune cu scop general. Un conector de alimentare auxiliar permite alimentarea separată a pinilor de alimentare asociați cu pinii D0-D13, ceea ce facilitează pentru acționarea servomotorelor. Există și un comutator de resetare la distanță pe scut pentru un acces ușor. De asemenea, ridică pe scut și pinul 13 - LED de pe placă pentru o vizibilitate mai bună [6].

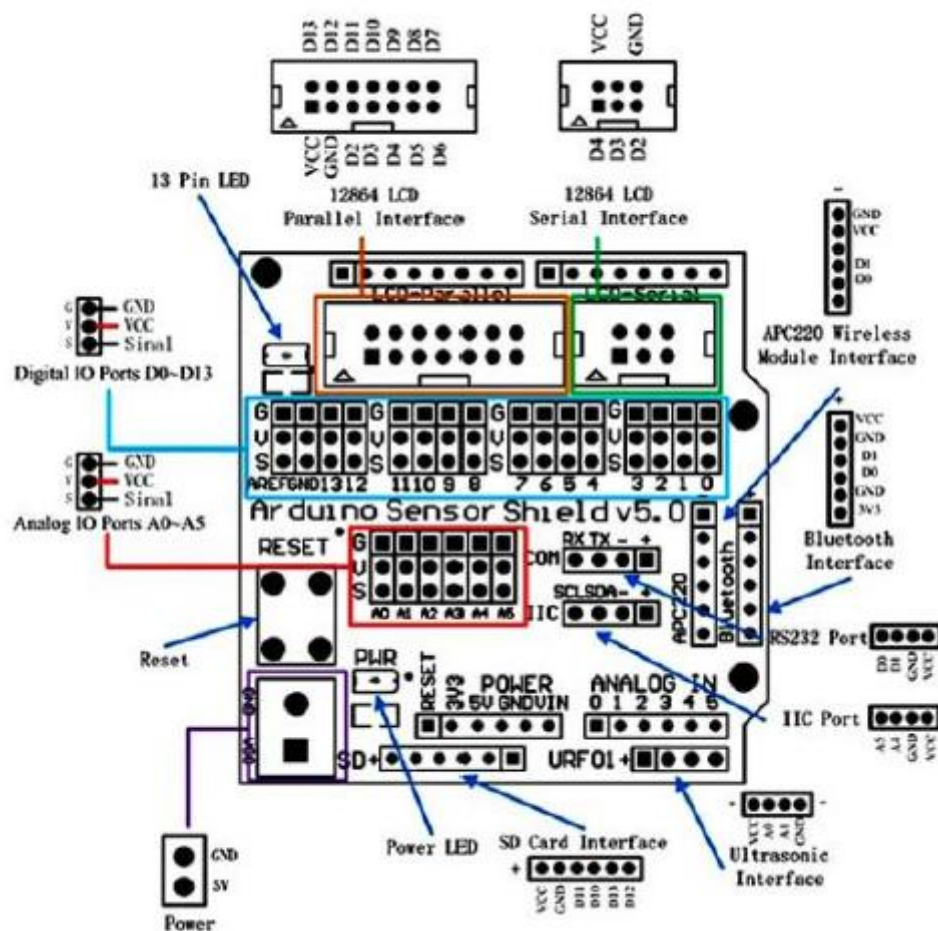


Fig. 5: Shield Sensor v5.0

3.2. Conexiunea Hardware

În figurile 6a și 6b se prezintă conexiunile folosite pentru realizarea proiectului. Breadboard-ul are rolul de Shield Sensor în diagrama făcută cu ajutorul Tinkercad. Cei mai importanți pini ai modului cu ultrasunete, Trig și Echo, sunt conectați la pinii digitali 10 și 11 ai lui Arduino. O observație importantă este că pinii 10 și 11 sunt utilizați de asemenea pentru conexiunile SPI, dar noi îi folosim pentru comunicarea cu modulul HC-SR04. Servomotorul este conectat la Arduino prin pinul 12, care este de asemenea un pin digital care poate fi utilizat și pentru comunicări SPI.

Senzorul cu ultrasunete emite un ultrasunet de 40 000 Hz care se deplasează prin aer și, dacă există un obiect sau un obstacol pe drumul său, revine înapoi la modul. Având în vedere timpul de călătorie și viteza sunetului putem calcula distanța parcursă. Pentru a genera ultrasunetele, trebuie să setăm Trig pe o stare HIGH timp de 10 μ s. Acest lucru va fi trimis o explozie sonică de 8 cicluri care va călători la viteza sunetului și va fi recepționată în pinul Echo. Pinul Echo va emite timpul, în microsecunde, parcurs de unda sonoră. Acest proces este redat în figura 7.

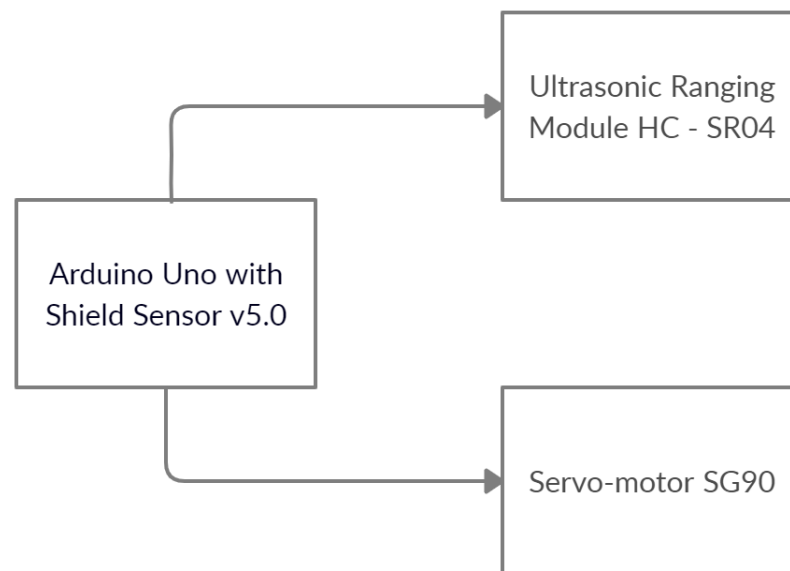


Fig. 6a: Design-ul hardware

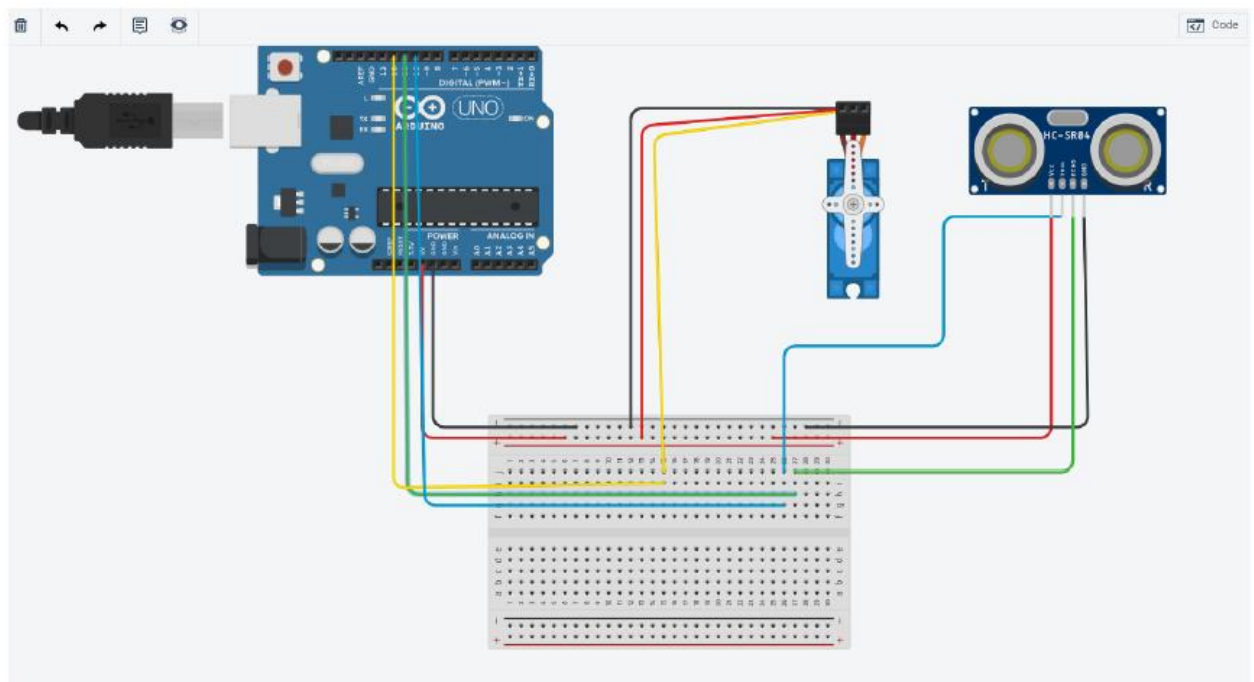


Fig. 6b: Conexiunile hardware

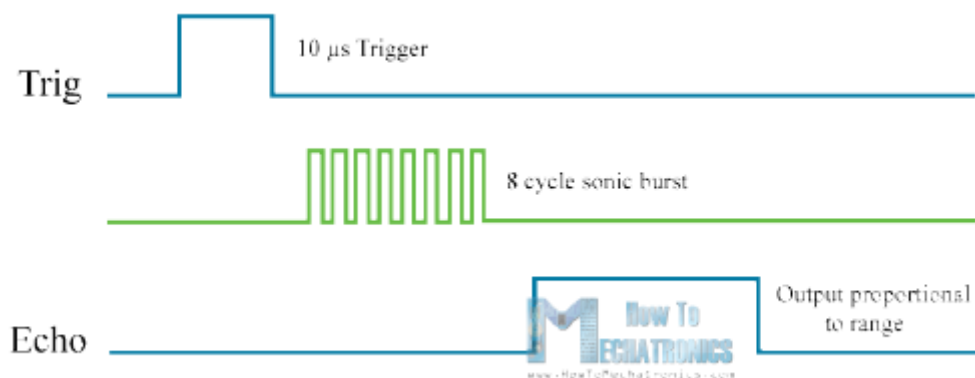


Fig. 7: Semnalele sonore [7]

4. Utilizare

4.1. Teste preliminare

Pentru a construi radarul vom efectua câteva teste pentru a găsi precizia senzorului folosit în proiect la diferite distanțe. Am pus un cod simplu Arduino pe placă doar pentru a măsura distanța față de un obiect plasat în fața senzorului la o anumită distanță. Realizăm mai multe teste pentru a vedea dacă distanța reală corespunde distanței tipărite pe monitorul serial. Analizând graficul prezentat în figura 8 putem vedea că senzorul nu are o precizie foarte bună, uneori eroarea relativă fiind mai mare de 20/100. De asemenea, se observă că eroarea crește direct proporțional cu creșterea distanței dintre obiect și senzor. Cauzele posibile ale acestor erori mari la distanțe mari pot fi refracțiile multiple ale undei de la senzor pe alte obiecte care nu au fost presupuse spre a fi măsurate, o calibrare greșită a senzorului sau faptul că modulul cu ultrasunete nu este perfect paralel cu solul. Experimentele într-un laborator sau folosind un giroscop pot crește performanța senzorului.

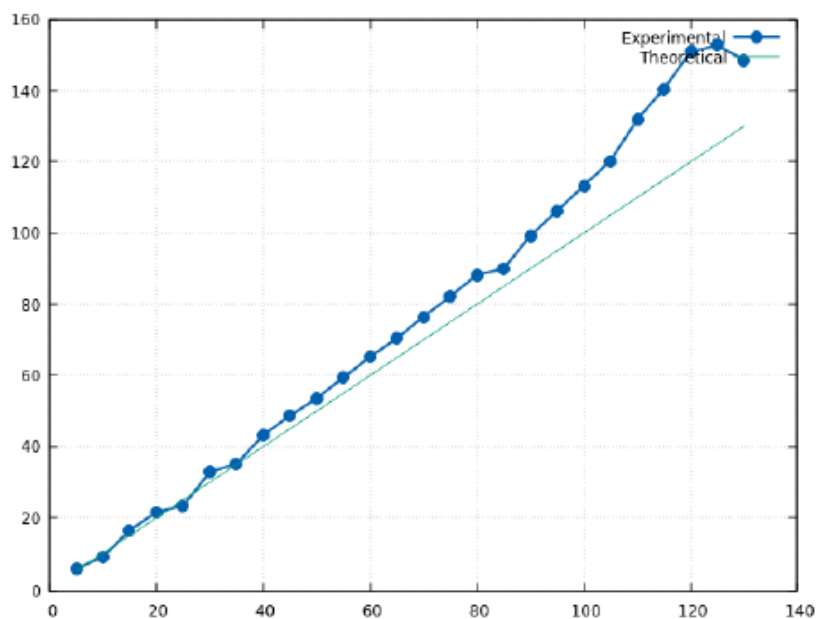


Fig. 8: Distanța teoretică [cm] vs. distanța măsurată [cm]

Utilizând o bibliotecă Arduino putem controla foarte ușor servomotorul folosind funcția „myServo.write(i)” și introducând o întârziere „delay(j)” înainte să folosim următoarea dată funcția „myServo.write(i + 1)” la un unghi diferit. La fiecare unghi vom calcula distanța față de un obiect ceea ce va introduce și o întârziere delay k. Deci întârzierea dintre deplasarea de la unghiul i la unghiul (i + 1) va fi (j + k).

4.2. Utilizarea aplicației radar

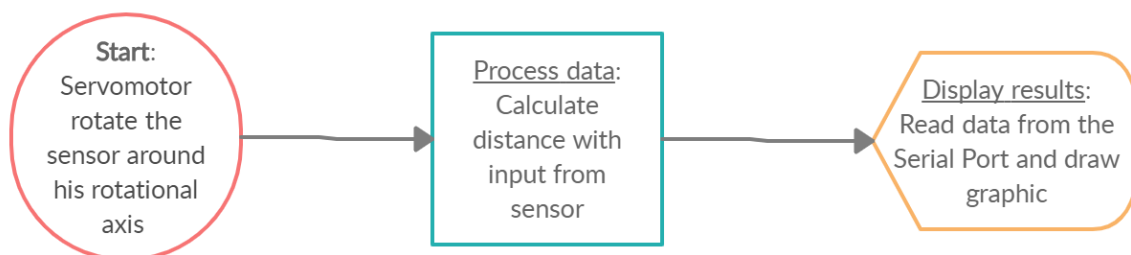


Fig. 9: Descrierea procesului

În figura 9 se prezintă procesul traversat de aplicația noastră pentru a ajunge la rezultatul final, afișarea distanței pe un grafic.

După ce conexiunile au fost făcute, codul Arduino și codul sursă sunt urcat pe placă. [8] Apoi se plasează un obiect în fața radarului. Servomotorul învârtă senzorul până când ultrasunetele emise de modulul de ultrasunete întâlnesc obiectul plasat. Folosind informația receptată și utilizând algoritmul din codul sursă, distanța până la obiect este calculată în cm și trimisă codului sursă prin portul serial. Codul sursă procesează informația primită de la Arduino și realizează interfața grafică, care este afișată pe ecran.

5. Rezultate și concluzii

5.1. Rezultate obținute în urma testării

Vom prezenta câteva dintre măsurătorile realizate. Range-ul nostru a fost setat între 0 cm și 40 cm. Obiectul a fost plasat la diferite distanțe.

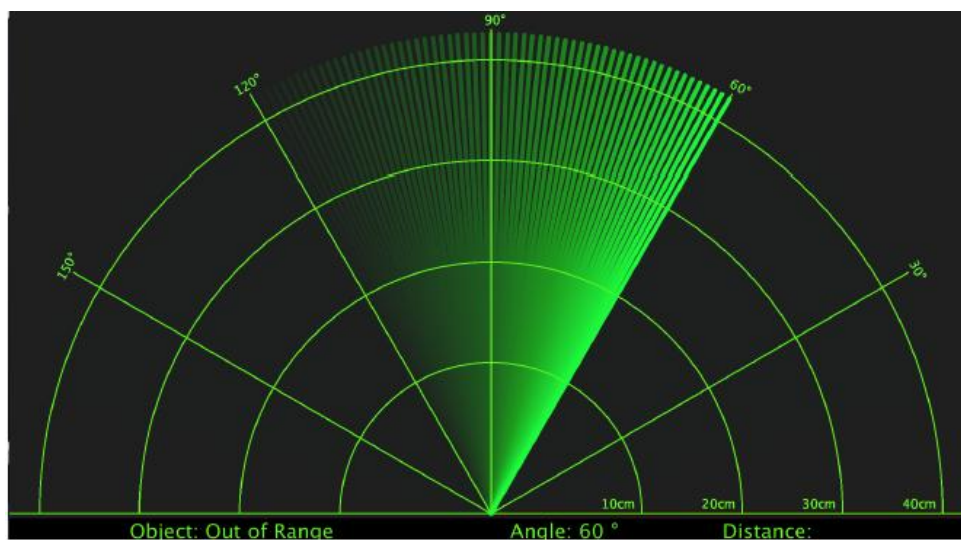


Fig. 10: Senzorul nu are niciun obiect în proximitatea lui

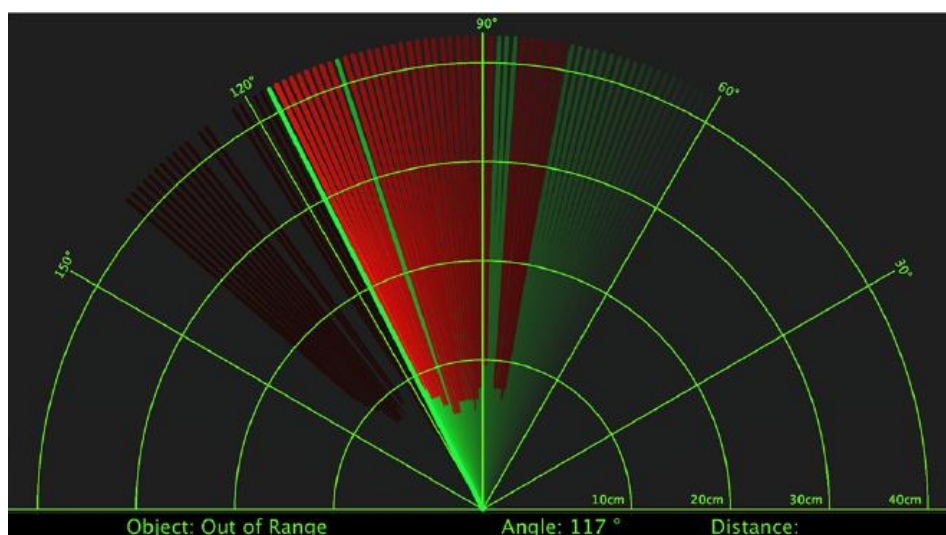


Fig. 11: Un obiect a fost detectat la o distanță foarte mică

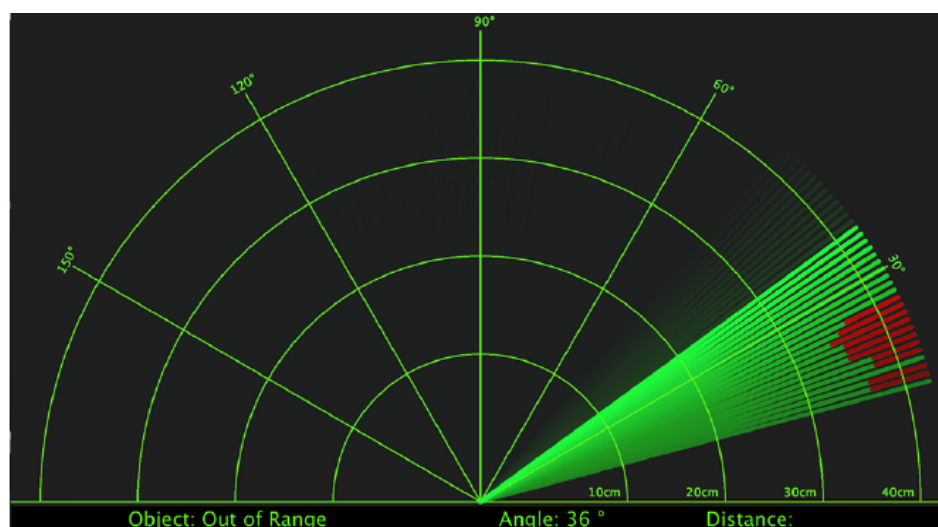
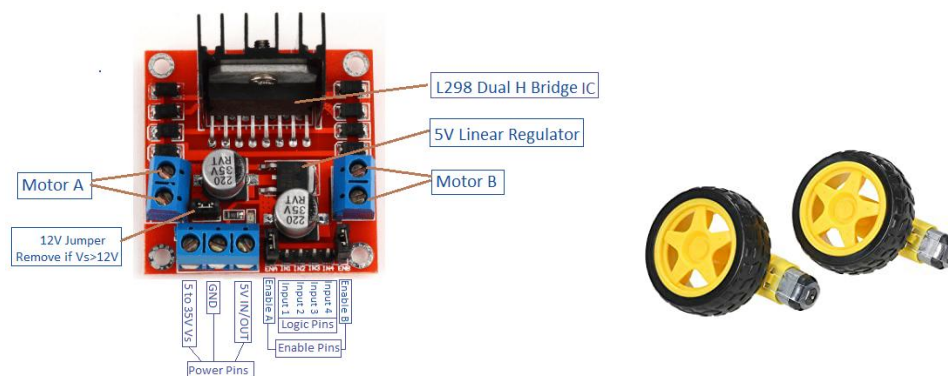


Fig. 12: Experimental obiectul se află la 35 cm; Chiar dacă radarul îl detectează la o distanță de aproximativ 36 cm, eroarea este mică

5.2. Optimizări posibile

Pentru a dezvolta proiectul, putem construi un robot Arduino care să evite obstacolele, care poate detecta obiectele din apropiere și poate să le evite. Este nevoie hardware suplimentar pentru ca noul prototip să mute sursa curentă, care nu va mai fi un PC conectat prin USB la placa Arduino. De asemenea, avem nevoie de o baterie de 9V conectată permanent la robot. Hardware-ul adițional este:

- Modulul L298N – conectat direct la placa Arduino pentru a controla roțile
- Roți electrice/ Motoare



5.3. Concluzii

Prin realizarea acestui proiect am învățat să folosim Processing IDE și ne-am îmbunătățit cunoștințele de folosire a Arduino. Partea dificilă a fost cea de montare, de realizare a conexiunilor hardware potrivite, dar și unele părți din cod. Chiar dacă a fost mai solicitantă intelectual, partea de montare a robotului a fost și cea mai plăcută și satisfăcătoare. Pe de altă parte, lucrul cu Processing IDE nu ne-a plăcut. În concluzie, cursul și laboratorul de CHS ne-au atins așteptările prin faptul că am învățat lucruri noi și am lucrat în echipă.

6. Referințe

- [1] - Marine radar, https://en.wikipedia.org/wiki/Marine_radar.
- [2] - Airport surveillance radar, https://en.wikipedia.org/wiki/Airport_surveillance_radar.
- [3] - Arduino, <https://ro.wikipedia.org/wiki/Arduino>.
- [4] - Ultrasonic Ranging Module HC - SR04, <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>.
- [5] - Prottoy, https://pr0tt0y.business.site/posts/1384101302503350709?fbclid=IwAR2aZYSi48HkeeWvFjm5ZV4qrtuZvhLMf66nsx1GvOB3ed9_42TsnvMOBvg.
- [6] - Proto Supplies, <https://protosupplies.com/product/sensor-shield-v5-0/>.
- [7] - How to Mechatronics, <https://howtomechatronics.com/>.
- [8] - Processing, <https://processing.org/reference/>.