

UNIVERSITATEA POLITEHNICA TIMISOARA
FACULTATEA DE AUTOMATICA SI CALCULATOARE
SECTIA CALCULATOARE SI TEHNOLOGIA INFORMATIEI

Sonar Radar
PROIECT REALIZAT CU ARDUINO

Coordonator:
IONASCU MARIAN- EMANUEL

Studenti:
SPATACEAN OANA- TEODORA
STOICHESCU IULIA

CUPRINS

1. Introducere
 - 1.1 Scopul Proiectului
 - 1.2 Problema Rezolvata
 - 1.3 Solutia Tehnica
2. Materiale și Componente
 - 2.1 Placa Arduino (Arduino UNO)
 - 2.2 Senzor Ultrasonic (HC-SR04)
 - 2.3 Servo Motor
 - 2.4 Breadboard și Fire de Conectare
3. Schema de Conectare
 - 3.1 Diagrama Electrică
 - 3.2 Descrierea Conexiunilor
4. Codul Sursă
 - 4.1 Explicația Codului
 - 4.2 Codul Integral
 - 4.3 Modul de Încărcare a Codului pe Arduino
5. Librării Utilizate
 - 5.1 Librăria Servo.h
 - 5.2 Librăria processing.serial
6. Funcționarea Proiectului
 - 6.1 Interfata de comunicare
 - 6.2 Module interne ale microcontrolerului Arduino
7. Concluzii
 - 7.1 Concluzii Generale
 - 7.2 Bibliografie

1. Introducere

● 1.1 Scopul Proiectului

Proiectul nostru se concentrează pe dezvoltarea unei aplicații Sonar/Radar utilizând un senzor ultrasonic și un servomotor, însoțită de o interfață grafică pentru a cartografia în timp real zona din jurul senzorului. Scopul este de a oferi utilizatorilor o modalitate eficientă de detectare a obiectelor din apropiere și de vizualizare a acestora într-un format ușor de înțeles.

● 1.2 Problema Rezolvata

În medii aglomerate sau în locații cu vizibilitate redusă, este crucial să identifiți obiectele din jur pentru a evita coliziunile sau pentru a naviga în condiții de siguranță. Proiectul nostru vine în întâmpinarea acestei nevoi, oferind o soluție accesibilă și eficientă pentru monitorizarea mediului înconjurător.

● 1.3 Solutia Tehnica

Utilizăm un senzor ultrasonic montat pe un servomotor controlat de un microcontroller pentru a efectua o baleiere continuă a zonei de vizibilitate. Senzorul măsoară în permanență distanța până la obiectele din jur, iar aceste date sunt transmise către o aplicație grafică care rulează pe un laptop, computer desktop, browser web sau dispozitiv mobil. Aplicația afișează în timp real o cartografiere a zonei, marcând cu verde zonele libere și cu roșu zonele în care sunt detectate obiecte, indicând și valoarea unghiului sub care sunt detectate acestea.

2. Materiale și Componente

2.1 Arduino UNO

Arduino UNO este o placă de dezvoltare microcontroller bazată pe ATmega328P. Este una dintre cele mai populare plăci din gama Arduino datorită ușurinței în utilizare și a comunității extinse de utilizatori și dezvoltatori. Caracteristici principale:

- Microcontroller: ATmega328P
- Tensiune de operare: 5V
- Tensiune de alimentare (recomandată): 7-12V
- Pini digitali I/O: 14 (dintre care 6 PWM)
- Pini de intrare analogică: 6
- Memorie Flash: 32 KB (din care 0.5 KB folosiți de bootloader)
- RAM: 2 KB
- EEPROM: 1 KB
- Viteză de ceas: 16 MHz

Arduino UNO este utilizat pentru a controla și monitoriza toate componentele din acest proiect. Este responsabil pentru citirea datelor de la senzorul ultrasonic și pentru controlul servo motorului.

2.2 Senzor Ultrasonic - HC-SR04

Senzorul ultrasonic HC-SR04 este utilizat pentru măsurarea distanței dintre senzor și un obiect. Acesta funcționează pe principiul ecoului: trimite un semnal ultrasonic și măsoară timpul până la recepția semnalului reflectat. Caracteristici principale:

- Tensiune de operare: 5V
- Curent de operare: < 2mA
- Unghi de măsurare: Aproximativ 15 grade
- Distanța de măsurare: 2 cm până la 400 cm
- Precizie: ± 3 mm
- Pini: VCC, Trig, Echo, GND

În proiect, HC-SR04 este folosit pentru a detecta obstacole și a măsura distanța până la acestea, informații care sunt utilizate pentru a crea o hartă radar.

2.3 SG90 Micro-servo motor

SG90 este un servo motor de dimensiuni mici, cunoscut pentru precizia și răspunsul său rapid. Este adesea utilizat în proiecte de robotică și automatizări datorită costului redus și ușurinței în control. Caracteristici principale:

- Tensiune de operare: 4.8V – 6V
- Cuplu: 1.8 kg·cm (4.8V)
- Viteză: 0.1 s/60° (4.8V)
- Unghi de rotație: 180 de grade (90° în fiecare direcție)
- Dimensiuni: 22.2 × 11.8 × 31 mm
- Greutate: 9 g

În proiect, SG90 este utilizat pentru a roti senzorul ultrasonic într-un arc de cerc, permițând măsurarea distanțelor în diferite direcții pentru a simula funcționalitatea unui radar.

● 2.4 Breadboard și Fire de Conectare

Breadboard-ul, cunoscut și sub numele de placă de testare, este un instrument esențial pentru prototiparea circuitelor electronice. Acesta permite realizarea rapidă și ușoară a conexiunilor între diferitele componente electronice fără a necesita lipirea acestora. Caracteristici principale:

- Dimensiuni tipice: 830 puncte de conexiune (sau similar)
- Dispunerea punctelor: Rânduri și coloane organizate într-un mod standardizat pentru conectarea componentelor
- Piste de alimentare: Linii orizontale pe marginea breadboard-ului pentru conectarea alimentării (+ și -)
- Material: Plastic ABS cu contacte din metal conductiv

Breadboard-ul este utilizat în proiect pentru a aranja și conecta componentele electronice (Arduino UNO, senzor ultrasonic, servo motor) într-o manieră organizată și flexibilă, permițând modificări rapide și testarea facilă a circuitelor.

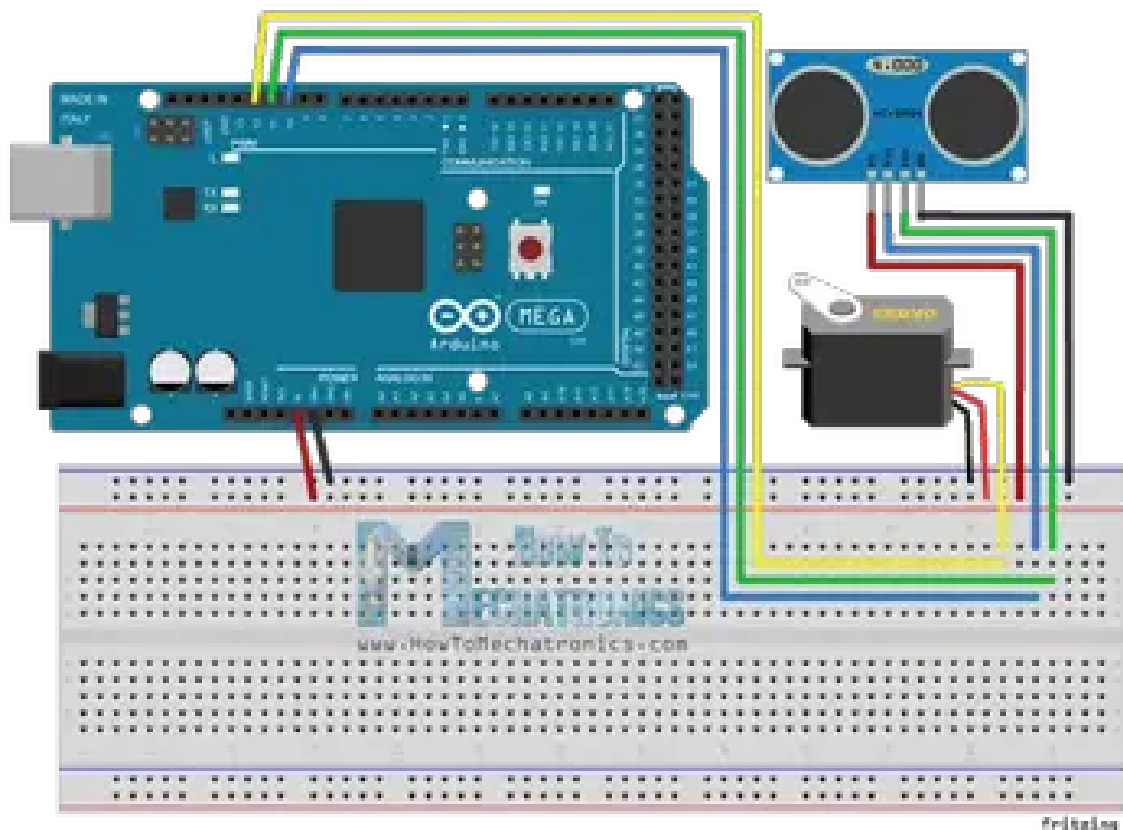
Firele de conectare (jumper wires) sunt utilizate pentru a realiza legăturile electrice între componentele montate pe breadboard și placa Arduino. Acestea vin în mai multe tipuri, dar cele mai comune pentru breadboard sunt firele male-to-male, male-to-female și female-to-female. Caracteristici principale:

- Tipuri de fire: Male-to-Male, Male-to-Female, Female-to-Female
- Lungimi variabile: Disponibile în diferite lungimi pentru a se potrivi nevoilor proiectului
- Culori multiple: Disponibile într-o gamă variată de culori pentru a facilita identificarea conexiunilor
- Conectori: Pin conectori standard care se potrivesc perfect în punctele de conexiune ale breadboard-ului și în header-urile Arduino

Firele de conectare sunt esențiale în acest proiect pentru a lega senzorul ultrasonic și servo motorul la pinii corespunzatori ai plăcii Arduino, asigurând alimentarea și comunicarea corectă între componente.

3. Schema de Conectare

• 3.1 Diagrama Electrică



● 3.2 Descrierea Conexiunilor

Senzor ultrasonic (HC-SR04)- Conexiuni

• VCC (fir roșu): Acesta furnizează energie senzorului. Este conectat la pinul de 5V pe Arduino, care furnizează 5 volți.

• GND (fir negru): Aceasta este conexiunea la masă pentru senzor. Este conectată la pinul GND pe Arduino.

• TRIG (fir galben): Acest pin declanșează unda ultrasonică. Este conectat la pinul digital 9 pe Arduino.

• ECHO (fir verde): Acest pin primește ecoul reflectat de obiect. Este conectat la pinul digital 10 pe Arduino.

Servo-motor - Conexiuni

• VCC (fir roșu): Acesta furnizează energie motorului servo. Este conectat la pinul de 5V pe Arduino.

• GND (fir negru): Aceasta este conexiunea la masă pentru motorul servo. Este conectată la pinul GND pe Arduino.

• Semnal (fir galben): Acesta transportă semnalul de control pentru motorul servo. Este conectat la pinul digital 11 pe Arduino.

Șine de alimentare ale breadboard-ului

Breadboard-ul are două rânduri lungi (șine) pentru distribuția alimentării:

Șina pozitivă (Roșu): Conectată la pinul de 5V al Arduino pentru alimentarea componentelor (senzor și servo).

Șina negativă (Albastru): Conectată la pinul GND al Arduino pentru a furniza masă componentelor.

De ce aceste conexiuni?

Alimentare:

- Atât senzorul ultrasonic, cât și motorul servo necesită o sursă de alimentare de 5V, pe care Arduino o furnizează. Șinele de alimentare de pe breadboard facilitează distribuția.

Masă:

- Conexiunile la masă sunt necesare pentru a închide circuitele pentru ambele componente. Șina de masă ajută la distribuirea masei de la Arduino către ambele componente.

Semnale de control:

- Pini TRIG și ECHO de pe senzorul ultrasonic necesită pini digitali separați pe Arduino pentru trimiterea și primirea pulsurilor ultrasonice.
- Semnalul de control al motorului servo este alimentat de un pin digital separat pentru a controla unghiul său de rotație.

În această configurație:

1. Senzorul ultrasonic detectează obiecte folosind pinii 9 (TRIG) și 10 (ECHO).
2. Motorul servo rotește senzorul folosind pinul 11.
3. Pinii de 5V și GND de pe Arduino furnizează alimentare și masă pentru ambele componente prin șinele de alimentare ale breadboard-ului.

8. Codul Sursă

● 4.1 Explicația Codului

Codul nostru din fisierul cu extensia .ino controlează un servomotor și un senzor ultrasonic pentru a măsura distanțele în jurul unei arii. Servomotorul se rotește, iar senzorul ultrasonic detectează obiectele și măsoară distanța până la acestea, afișând rezultatele prin comunicare serială.

Datele colectate sunt trimise către un computer pentru afișare grafică în timp real folosind Processing. Codul nostru din fisierul cu extensia .pde este împărțit în mai multe funcții care se ocupă de citirea datelor, desenarea radarului, afișarea obiectelor detectate și a textului informativ.

● 4.2 Codul Integral

Codul integral se găsește la acest link de pe Github:
https://github.com/IuliaStoichescu/Sonar-Radar_Project

● 4.3 Modul de Încărcare a Codului pe Arduino



1. Deschideți Arduino IDE.
2. Copiați și lipiți codul de mai sus în editorul de text al Arduino IDE.
3. Conectați placa Arduino la computer folosind un cablu USB.
4. Selectați tipul plăcii și portul corespunzător din meniul "Tools".
5. Apăsați butonul "Upload" pentru a încărca codul pe Arduino.

5. Librării Utilizate

● 5.1 Librăria Servo.h

Librăria Servo.h, este specifică platformei Arduino și este folosită pentru controlul servomotoarelor. Această bibliotecă furnizează funcții și metode pentru a controla ușor și eficient poziția unui servomotor. De exemplu, poziția unui servomotor poate fi setată prin specificarea unghiului dorit în grade.

Această bibliotecă este compatibilă cu arhitecturile avr, megaavr, sam, samd, nrf52, stm32f4, mbed, mbed_nano, mbed_portenta, mbed_rp2040, renesas, renesas_portenta, renesas_uno, astfel încât să poată fi utilizată pe placa Arduino UNO pe care am folosit-o.

Biblioteca poate controla un număr mare de servo-motoare, făcând uz atent de temporizatoare: biblioteca poate controla 12 servo-motoare folosind doar 1 temporizator.

În ceea ce privește biblioteca Servo.h și modulul servo utilizat în cod, modul în care acestea funcționează este la un nivel mai detaliat.

1. Funcționarea bibliotecii Servo.h:

Această bibliotecă abstractizează interacțiunea cu modulul servo, permițându-ți să controlezi servo-motorul folosind un set simplu de funcții, cum ar fi attach(), write(), writeMicroseconds(), etc. De obicei, această bibliotecă utilizează temporizatoare interne ale microcontroller-ului pentru a genera semnale PWM care controlează poziția servo-motorului.

2. Modulul servo:

Modulul servo este o componentă care răspunde la semnalele PWM pentru a-și regla poziția. Protocolul standard pentru controlul servomotoarelor implică trimiterea unui semnal PWM cu o perioadă de 20 de milisecunde (ms) și un raport al datoriei între 1 și 2 ms. O lungime a impulsului de 1 ms indică o poziție complet închisă, în timp ce o lungime a impulsului de 2 ms indică o poziție complet deschisă. Lungimea impulsului între aceste valori reprezintă pozițiile intermediare ale servo-motorului.

3. Implementarea în codul nostru:

În codul nostru, utilizarea bibliotecii Servo.h este evidentă prin includerea `#include<Servo.h>` și prin inițializarea obiectului Servo cu `Servo myservo;`. Apoi, prin utilizarea funcției `myservo.write(i)`. În buclele `for`, setăm unghiul servo-motorului la valoarea `i`, ceea ce implică generarea unui semnal PWM corespunzător pentru a controla poziția servo-motorului.

Biblioteca Servo.h și modulul servo funcționează împreună pentru a permite controlul precis al poziției servo-motorului prin generarea de semnale PWM, ceea ce este esențial pentru proiectul nostru de rotire și măsurare a distanței cu senzorul ultrasonic.

● 5.2 **Librăria `processing.serial`**

Libraria `processing.serial` este utilizată în medii de programare cum ar fi Processing pentru a permite comunicarea serială între calculator și alte dispozitive, cum ar fi plăcile de dezvoltare Arduino. Această bibliotecă oferă funcții pentru a inițializa, citi și scrie date prin portul serial.

Biblioteca Serial citește și scrie date de la și către dispozitive externe câte un byte la un moment dat. Permite a două calculatoare să trimită și să primească date. Această bibliotecă oferă flexibilitatea de a comunica cu dispozitive microcontroler personalizate și de a le folosi ca intrare sau ieșire pentru programele Processing. Portul serial este un port I/O cu nouă pini care există pe multe PC-uri și poate fi emulat prin USB.

Atunci când se trimit date către consolă, cum ar fi prin `print()` sau `println()`, consola este relativ lentă. Libraria nu suportă ieșirea în timp real la viteză mare (cum ar fi la 60 de cadre pe secundă). Pentru monitorizarea în timp real a valorilor seriale, se afișează acele valori în fereastra Processing în timpul `draw()`.

În codul nostru, din lista de metode și funcții ale bibliotecii Serial, sunt folosite următoarele:

- `import processing.serial.*`; Această linie importă biblioteca necesară pentru comunicarea serială.
- `serialPort = new Serial(this, "COM4", 9600)`; Inițializează un obiect Serial pentru comunicarea cu un dispozitiv conectat la portul COM4 la o viteză de transfer de 9600 de biți pe secundă.
- `serialPort.bufferUntil('.')`; Setează un caracter specific ('.') pentru a fi păstrat în buffer până la momentul apelului `serialEvent()`.
- `serialEvent(Serial serialPort)`: Această funcție este apelată atunci când sunt disponibile date pe portul serial. În cadrul acestei funcții, se citesc datele de la portul serial și se interpretează unghiul și distanța obiectului detectat.

Ele sunt esențiale pentru comunicarea serială cu dispozitivul conectat la portul COM și pentru interpretarea datelor primite.

6. Funcționarea Proiectului

● 6.1 Interfata de comunicare

Interfața de comunicare este realizată prin utilizarea protocolului Serial între Arduino și calculatorul gazdă.

1. Serial.begin(9600):

În funcția `setup()`, se inițializează comunicarea serială la o rată de transfer de 9600 de biți pe secundă între Arduino și calculator.

2. Serial.print():

În funcția `loop()`, valorile unghiului și distanța măsurată sunt trimise pe portul serial către calculator utilizând funcția `Serial.print()`. Acest lucru se întâmplă pentru fiecare iterație a buclei `for` care rotește servomotorul într-o direcție și apoi în cealaltă.

Interfața de comunicare serială permite Arduino să transmită date către calculator, astfel încât acestea să poată fi afișate sau procesate ulterior. În acest caz, valorile unghiului și distanța măsurată sunt trimise pentru a fi afișate pe calculator. Utilizatorul poate citi și interpreta aceste date pentru a înțelege cum se comportă sistemul în timpul funcționării.

● 6.2 Module interne ale microcontrolerului Arduino



Timerul (Timer):

Deși nu este utilizat explicit în codul nostru, funcția `delay()` implică utilizarea temporizatorului intern pentru a suspenda execuția programului pentru o anumită perioadă de timp. Timerul poate fi, de asemenea, utilizat pentru a genera semnale de PWM (modularea lățimii impulsului) pentru controlul precis al servo-motorului, dar aceasta este gestionată implicit de biblioteca `Servo.h`.

Convertor analog-digital (ADC):

Nu este utilizat explicit în cod, deoarece senzorul utilizat este un senzor ultrasonic care utilizează un protocol digital pentru măsurători. Dacă aveam un senzor analog, am fi putut folosi modulul ADC pentru a converti semnalul analogic în valori digitale pe care Arduino le poate citi și procesa.

Porturile de intrare/ieșire (I/O):

Acestea sunt utilizate în mod activ pentru a controla pinii Arduino. Pinii `trigPin` și `echoPin` sunt configurați ca ieșire și intrare, respectiv, pentru comunicarea cu senzorul ultrasonic. De asemenea, pinul 12 este folosit pentru a controla servo-motorul.

Modularea lățimii impulsului (PWM):

Acesta este folosit pentru a controla servo-motorul. Biblioteca `Servo.h` utilizează unul dintre timerele interne ale Arduino pentru a genera semnale PWM precise care controlează poziția servo-motorului.

Am utilizat mai multe module interne ale microcontrolerului Arduino pentru a controla și a interacționa cu diferite componente hardware, precum senzorul ultrasonic și servo-motorul. Utilizarea acestor module este gestionată în mare parte de biblioteci precum Servo.h, care abstractizează interacțiunea cu aceste module, permițând programatorului să se concentreze pe logica aplicației.

7. Concluzii

• 7.1 Concluzii Generale

Proiectul nostru utilizează un senzor ultrasonic montat pe un servomotor pentru a efectua o scanare continuă a mediului înconjurător. Datele de distanță sunt transmise către o aplicație grafică care cartografiază în timp real zona din jur, marcând obiectele detectate. Oferim o soluție practică și eficientă pentru detectarea și navigarea în medii aglomerate sau cu vizibilitate redusă.

• 7.2 Bibliografie

<https://www.arduino.cc/reference/en/libraries/servo/>
http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf
<https://processing.org/reference/libraries/serial/index.html>
<https://www.sciencedirect.com/topics/engineering/breadboard>
<https://www.circuito.io/blog/arduino-code/>
https://www.google.com/search?sca_esv=acc4d604aee1a7ed&rlz=1C1GCEU_enRO1083RO1083&sxsrf=ADLYWIKHQVBj2_MYvjVwHTl8SEo0PnMTIQ:1716044835238&q=sonar+radar+arduino+electric+scheme&uds=ADvngMj3w8HS8EBNkGbNHQjAWXe_OpQRgp_9KBJ9Cr-im0HrNvosqp1Ip3W-AeBRO6PmC65ZQbIlI9rKdpWEVVAmnCGen6y5zA-P8aC0c91M8udA3nzbEEbTWWc11CUiLib8LE4fNSdJM518j6ZVA6OIE7vE3Sptl3zRwC3iziDDN2b8d8TH-2d7M08kGQPnzzKNhzeet7coxNnFdq_WjotshQzp5JQSxsAUpB8320oLrQg6JdwyMW0VVK63yGYBr6Un9NRrxirU9nFToZPd2Dtsjm6k7hmSn4GMxAkpKbOYI5IzLUEx_WY9HIVJyG-SD3Z0FUw4xgQj&udm=2&prmd=ivnbz&sa=X&ved=2ahUKEwjCpsqWvZeGAxXfgP0HHXr8AnkQtKgLegQIDBAB&biw=1536&bih=703&dpr=1.25#vhid=tnaP-dua3nA9ZM&vssid=mosaic