



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

Robot pentru detecția obstacolelor cu posibilitate de control prin Bluetooth

Nume: Chiorean Bogdan Emanuel

Hădărău Elena

Grupa: 30233

Introducere

În era tehnologiei moderne, proiectele de robotică reprezintă o modalitate captivantă de a explora și de a înțelege principiile fundamentale ale programării și a sistemelor automate. Proiectul propus aici îmbină elemente de robotică și tehnologie wireless, oferind o experiență interactivă și educativă. Este vorba despre un robot cu Arduino echipat cu un sistem de detecție a obstacolelor, ce permite controlul său prin intermediul tehnologiei Bluetooth.

Robotul va fi programat să efectueze scanări constante ale mediului său cu ajutorul senzorului de distanță. În cazul detectării unui obstacol, în funcție de modul selectat de acțiune, robotul va evita obstacolele în mod diferit. Mai exact, dacă este în modul autonom, algoritmul de evitare va interveni pentru a schimba direcția robotului și a evita coliziunea. În caz contrar, dacă este în modul manual, prin intermediul modulului Bluetooth, utilizatorul va putea prelua controlul și să direcționeze robotul care se află în stare de repaus până ce își primește comanda.

Astfel, obiectivele proiectului sunt: înțelegerea și aplicarea principiilor de bază ale programării în limbajul C/C++, dar în mediul de dezvoltare Arduino IDE; experiența practică cu senzori și motoare într-un context robotic, dezvoltarea cunoștințelor în domeniul comunicării wireless prin Bluetooth și explorarea conceptelor de autonomie și control manual în robotică.

Studiu bibliografic

- Senzor Ultrasonic HC-SR04

Senzorul ultrasonic HC-SR04 este un dispozitiv utilizat pentru măsurarea distanței folosind unde sonore ultrasonice. Emite impulsuri sonore, măsoară timpul până la revenirea ecoului și calculează distanța. Este compus dintr-un transmițător (Trig) și un receptor (Echo), se conectează ușor la microcontrolere precum Arduino și este frecvent utilizat în proiecte de robotică pentru detecția obstacolelor și navigarea autonomă.

Tensiune de alimentare: 5V

Distanță detectată: 2 cm ~ 450 cm

Precizie: 0.3 cm

- Motoare DC

Motoarele de curent continuu (DC motors) clasice convertesc energia electrică în lucru mecanic. Viteza de rotație a unui motor este proporțională cu tensiunea de alimentare de la bornele acestuia, iar direcția de rotație depinde de polaritate (conectarea celor 2 fire de alimentare ale motorului la +Vcc și Gnd, sau vice-versa). Motoarele au cutie de viteze (reductor de turație) cu raport de 1:48, ceea ce înseamnă că pentru o rotație completă a axului extern se efectuează de fapt 48 de rotații ale motorului electric. Folosirea unui reductor are avantajul că mărește forța de acționare, cu costul vitezei.

Tensiune de alimentare motor: 3V - 6V

- Puntea H Dublă L298N

Puntea duală L298N este un circuit integrat utilizat pentru controlul motoarelor de curent continuu (DC) în aplicații precum robotică și proiecte DIY. Cu două punți H-bridge independente, L298N permite controlul direcției și vitezei a două motoare simultan. Este potrivit pentru proiecte care necesită manipularea precisă a mișcării, cum ar fi roboții autonomi sau vehiculele teledirijate. L298N funcționează cu semnale logice pentru a determina direcția și viteza motorului și poate fi integrat eficient cu platforme de dezvoltare, cum ar fi Arduino.

- Servo-Motor

MicroServo 9g SG90 este un mic motor servo cu o greutate de aproximativ 9 grame, utilizat în proiecte de robotică. Având dimensiuni compacte, acest servo oferă un unghi de rotație limitat de obicei la 180 de grade și funcționează la o tensiune de 4,8V - 6V. Motoarele servo sunt folosite pentru a obține rotații parțiale, stabile și controlate, pentru efectuarea unor operații cu amplitudine mică dar cu precizie ridicată: acționare mecanism de închidere-deschidere, poziționare senzori, efectuarea unor gesturi, etc. Cel mai simplu mod de a lucra cu aceste motoare este prin folosirea bibliotecii Servo.

Motoarele servo au 3 fire. Firul de culoare roșie - Vcc (5V), cel maro - GND. Pe lângă aceste două fire de alimentare, există un al treilea, firul de comandă, cel portocaliu. În funcție de producător culorile pot varia.

- Modulul Bluetooth HC-05

Modulul Bluetooth HC-05 este un dispozitiv transceiver serial Bluetooth care consumă foarte puțină energie, proiectat pentru a facilita comunicația wireless între dispozitive. Cu o funcționalitate Bluetooth clasică, modulul operează ca un transceiver serial, permitând transmiterea și recepționarea datelor între dispozitive prin porturile TX și RX. Este utilizat în proiecte de robotică, pentru controlul la distanță al dispozitivelor prin Bluetooth și se integrează bine cu platforme de dezvoltare precum Arduino.

Tensiune de alimentare: 3.6 - 6V;

Comunica pe serial UART;

Baudrate: 9600 - 460800 bps;

Distanța de transmisie până la 10m;

Putere de transmisie: +4dBm;

Senzitivitate recepție: -80dBm.

- Ecranul LCD 1602 verde cu modul IIC

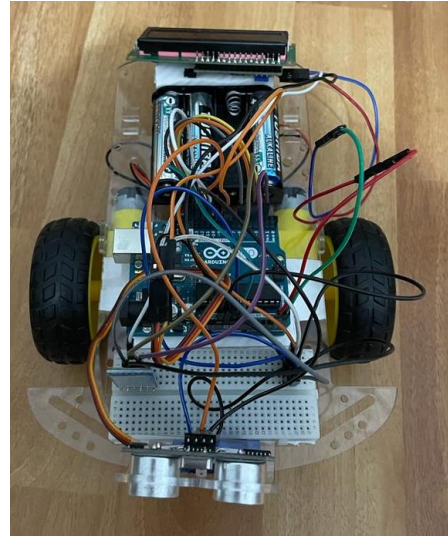
Ecranul LCD 1602 verde cu modul IIC este un afișaj cu cristale lichide de 16 caractere pe două linii, iluminat cu verde, și include un modul de comunicație IIC (Inter-Integrated Circuit). Acesta simplifică conexiunea cu dispozitivele prin intermediul a doar câtorva pini, facilitând integrarea în proiecte. Permite ajustarea contrastului și luminozității și este compatibil cu diverse platforme de dezvoltare precum Arduino și Raspberry Pi. Utilizat în aplicații variate, oferă o modalitate eficientă de afișare a informațiilor în proiecte de robotică.

Pentru realizarea proiectului, am lucrat în mediul de dezvoltare Arduino IDE. Limbajul de programare utilizat în Arduino IDE este bazat pe C/C++ și este adaptat pentru a simplifica programarea pentru utilizatorii începători.

Design

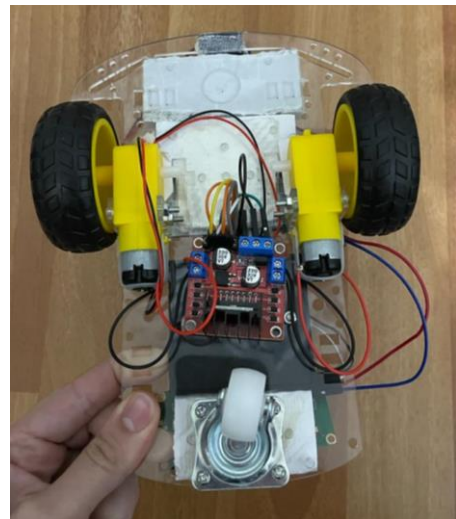
Vedere de sus:

1. Placă Arduino Uno
2. O placă de prototipizare (Breadboard)
3. Senzor Ultrasonic
4. Modulul Bluetooth
5. 1 Motor Servo
6. Carcasă baterii 4xAA (R6)
7. LCD 1602 verde cu modul IIC



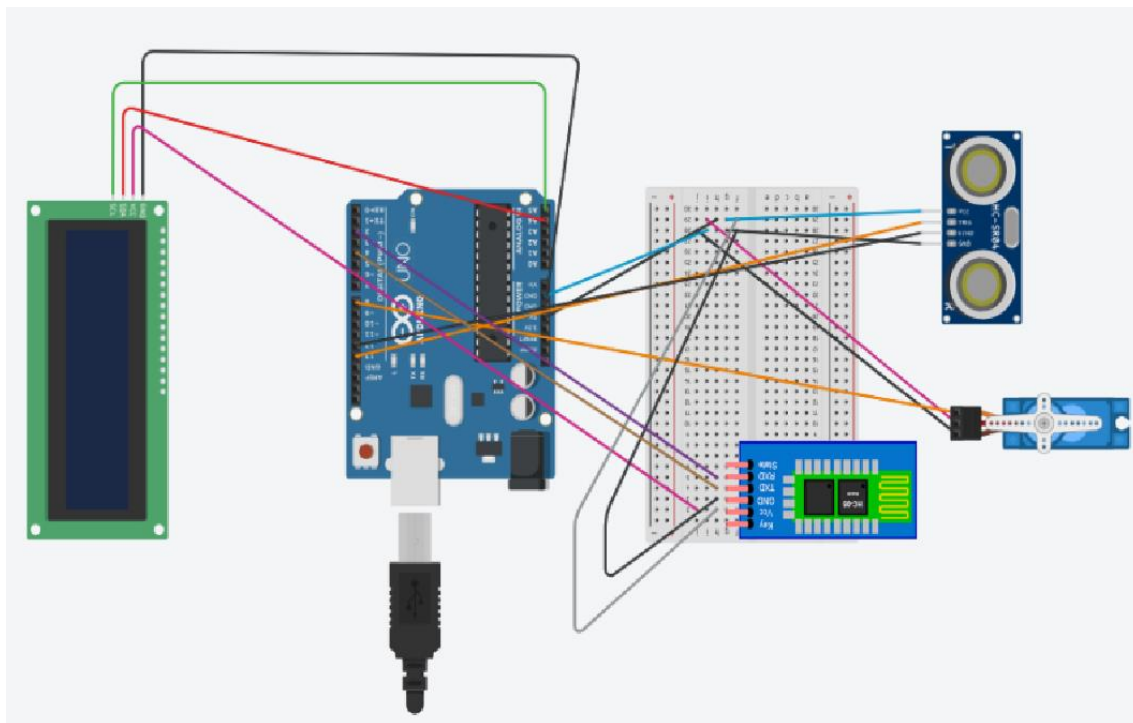
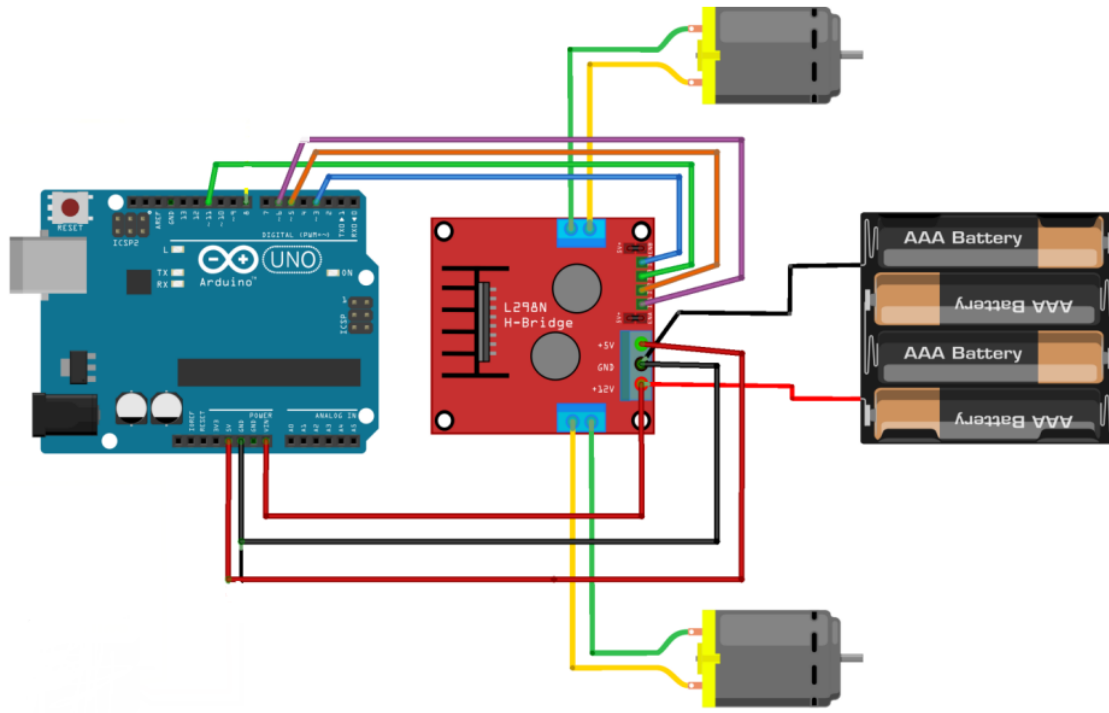
Vedere de jos:

1. Driver motoare L298N Dual H-Bridge
2. 2x Motor DC
3. 2 roți conectate la motoare, 1 roată suport
4. Suport plexiglas



Implementare

Schema:



Funcția de calculare a distanței: `getDistance()` – returnează distanța până la obstacol

```
long getDistance(){
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    // Convert the time into a distance
    return ((duration/2) / 29.1);    // Divide by 29.1 or multiply by 0.0343
}
```

Funcția de verificare a distanței: `checkDistance()` – returnează o valoare booleană: '1' – True: dacă cel puțin o distanță măsurată este mai mare de 20 de cm, însemnând că robotul se poate deplasa sau '0' – False: dacă cele 2 distanțe sunt mai mici și, în acest caz, robotul trebuie să se oprească.

```
bool checkDistance(){
    cm=getDistance();
    cm2=getDistance();
    bool ver=((cm>20) || (cm2>20));
    bt.print(ver);
    lcd.setCursor(6,0);
    if(ver){
        lcd.print("YES");
    }else
    {
        lcd.print("NO ");
    }
    return ver;
}
```

Funcția pentru controlul unui motor: void StartMotor (int m1, int m2, int forward, int speedDC) - m1 și m2 sunt pinii de control ai punții H care alimentează motorul, forward este o variabilă booleană care indică direcția mișcării motorului (true - motorul se va mișca înapoi, false - se va mișca înainte), speedDC este variabila care indică viteza motorului. Aceasta trebuie să fie o valoare între 0 și 255 și este controlată prin modularea în lățime a impulsului (PWM).

```
void StartMotor (int m1, int m2, int forward, int speedDC)
{
    if (speedDC==0) // oprire
    {
        digitalWrite(m1, 0);
        digitalWrite(m2, 0);
    }
    else
    {
        if (forward)
        {
            digitalWrite(m2, 0);
            analogWrite(m1, speedDC); // folosire PWM
        }
        else
        {
            digitalWrite(m1, 0);
            analogWrite(m2, speedDC);
        }
    }
}
```

Funcție de siguranță: void delayStopped(int ms) – execută oprire motoare, urmată de delay

```
void delayStopped(int ms)
{
    StartMotor (mpin00, mpin01, 0, 0);
    StartMotor (mpin10, mpin11, 0, 0);
    delay(ms);
}
```

Funcții de deplasare înainte și înapoi: void dirForward(), void dirBack()

```
void dirForward()
{
    // playWithServo(8,90);
    StartMotor (mpin00, mpin01, 0, speedDCL);
    StartMotor (mpin10, mpin11, 0, speedDCR);
}
void dirBack()
{
    // playWithServo(8,90);
    StartMotor (mpin00, mpin01, 1, speedDCL);
    StartMotor (mpin10, mpin11, 1, speedDCR);
}
```


Funcții de deplasare stânga, dreapta pentru modul manual:

```
void dirLeft()
{
    playWithServo(8,180);
    if(checkDistance()){
        // Acum se porneste doar un motor
        StartMotor (mpin10, mpin11, 0, speedDCR);
        delay(300);
    }
    playWithServo(8,90);
    delayStopped(400);
}
```

```
void dirRight()
{
    playWithServo(8,0);
    if(checkDistance()){
        // Acum se porneste doar un motor
        StartMotor (mpin00, mpin01, 0, speedDCL);
        delay(300);
    }
    playWithServo(8,90);
    delayStopped(400);
}
```

Funcții de deplasare stânga, dreapta pentru modul automat:

```
void dirLeftAuto()
{
    // Acum se porneste doar un motor
    StartMotor (mpin10, mpin11, 0, speedDCR);
    delay(300);
}

long distanceLeft()
{
    playWithServo(8,180);
    long dis= getDistance();
    playWithServo(8,90);
    delayStopped(300);
    bt.print("Left: ");
    bt.print(dis);
    bt.print(" ");
    lcd.setCursor(0,1);
    lcd.print("L:");
    lcd.print(dis);
    return dis;
}
```

```
void dirRightAuto()
{
    // Acum se porneste doar un motor
    StartMotor (mpin00, mpin01, 0, speedDCL);
    delay(300);
}

long distanceRight()
{
    playWithServo(8,0);
    long dis= getDistance();
    playWithServo(8,90);
    delayStopped(300);
    bt.print("Right: ");
    bt.print(dis);
    bt.print(" ");
    lcd.setCursor(8,1);
    lcd.print("R:");
    lcd.print(dis);
    return dis;
}
```

Probleme

Primele probleme au apărut în procesul de montare. Inițial am folosit la laboratoare robotul de la facultate, însă pierdeam mult timp (vreo 30 min per lab) să remontăm robotul, iar o dată l-am găsit defect (fire rupte, și a fost nevoie să schimbăm motorul). După ultimul laborator (deși am fost și la laboratoare extra cu alte grupe), ne-am achiziționat propriul robot cu aceeași structură pentru a termina și testa funcționalitățile.

Principală problemă pe care o avem este, faptul că, deși logica din cod pare a fi corectă, robotul la execuția comenzilor nu merge drept. Am măsurat rotațiile pe minut la ambele roți (am obținut aceleași valori pentru ambele roți) și am încercat să schimbăm voltajul, dar nu erau rezultate consistente. Concluzia la care am ajuns, după ce am încercat aceste metode de rezolvare și după vizualizarea unor videoclipuri cu roboți cu aceeași structură, este că a treia roată (cea cu rulmenți), nu funcționează conform așteptărilor: are greutate prea mare încât sunt apăsați rulmenții, iar roata se rotește greu. Soluția ar putea fi schimbarea celei de a treia roți sau schimbarea structurii robotului, însă după implementarea celorlalte funcționalități, e prea târziu.

Mod de funcționare

Pentru a ne putea conecta prin bluetooth la robot, am folosit aplicația Serial Bluetooth Terminal. Pentru conectare: dispozitivul se numește HC-05 și parola este: 1234. După conectarea cu succes, aplicația oferă o interfață unde se pot introduce comenzile pentru robot. Astfel, noi am codificat următoarele comenzi:

Pentru modul manual:

- f – pentru deplasare în față până întâlnește un obiect
- b – pentru deplasare cu spatele
- r – pentru rotire la dreapta și ulterior mers în față
- l – pentru rotire la stânga și ulterior mers în față

Pentru a interschimba între modul manual și cel autonom se folosește comanda: m