

Line Follower

Introducere

- Proiectul constă într-o mașinuță care urmărește traseul unei linii negre. Aceasta este capabilă să emită sunete, iar atunci când întâlnește un obstacol se oprește și emite un zgomot de claxon către obstacol, urmând să îl evite.
- Ideea a pornit de la curiozitatea mea de a face un proiect cât mai aproape de realitate, și anume, o mașină care scoate zgomote specifice, poate urmări un traseu și evita obstacolele.
- Ca utilitate, proiectul reprezintă un prim pas în înțelegerea procesului de dezvoltare a unei mașini autonome, capabile să funcționeze fără șofer.

Descriere generală

Proiectul constă într-o mașină care funcționează pe baterii. Aceasta are 2 roți mari în față și una singură în spate ca să poată face viraje. Cu ajutorul senzorului ultrasonic, depistează obstacolele, astfel oprindu-se la ele. De asemenea, la sesizarea unui obstacol, cu ajutorul unor sunete redade printr-un difuzor, mașina emite zgomot prin difuzor. Langa sol, se afla și senzorul de urmarire al liniilor.



Hardware Design

Lista de componente:

- Arduino UNO Atmega 328p
- breadboard
- senzor urmarire linie TCRT5000
- senzor ultrasonic HC-SR04
- baterii
- fire
- difuzor
- modul Driver de Motoare Dual în Miniatură L298N
- 2 motoare
- modul liniar ridicator de tensiune SX1308



În acest design am pus 4 baterii de tip AAA, fiindcă aplicația în care am făcut design-ul nu mi-a permis altfel, însă schema bloc și proiectul au bateriile AA. Am făcut firele cu culori ca să se înțeleagă mai

bine care unde intra.

Software Design

Descriere: Pus pe un traseu format din dungii negre, robotul incepe sa mearga pe ruta acestora. La intalnirea unui obstacol, acesta se opreste si incepe sa emita sunete pana cand obstacolul este indepartat. Apoi, isi continua drumul pana la intalnirea altui obstacol.

Mediul de dezvoltare este Arduino IDE. Am folosit bibliotecile NewPing (pentru senzorul ultrasonic) si PCM (pentru difuzor). Mi-am definit functii pentru mersul inainte, mersul inpoi, stanga dreapta. In functie de valorile date de senzorul de urmarire al linilor (L, C si R) am creat un algoritm care decide cand este necesar sa mearga drept sau sa faca viraje. Sunetul redat pe difuzor l-am obtinut convertind un mp3 in wav, iar apoi intr-un vector de numere. Am definit toti pinii folositi, mai putin pinul 11 care este folosit automat de biblioteca PCM pentru redarea sunetului. De asemenea, am facut un fel de automat de stari, pentru a stii cand masina merge (MOTOR_ACTIV), cand difuzorul merge(SUNET_ACTIV), sau cand este intr-o stare intermediara (OPRIT). Difuzorul porneste atunci cand intampina un obstacol, iar motorul porneste atunci cand obstacolul este indepartat.

Ca laboratoare, am folosit urmatoarele: timere, intreruperi pentru redarea sunetului pe speaker, PWM pentru a controla viteza motoarelor. De asemenea, laboratorul de GPIO m-a ajutat sa pot face configurari hardware corecte: Pinii pinL, pinC, și pinR sunt configurați ca intrări. Acești pini sunt folosiți pentru a citi stările digitale. Pinii motorL1, motorL2, motorR1, și motorR2 sunt configurați ca ieșiri.. Această configurare permite controlul driverului de motor conectat la acești pini. Deoarece Arduino nu poate furniza curentul necesar direct motoarelor, utilizarea acestor pini ca ieșiri facilitează comanda semnalelor către circuitele externe care pot gestiona curentii mai mari necesari pentru operarea motoarelor. Un pin de la un Arduino poate furniza un curent maxim ce este insuficient pentru a alimenta direct motoarele, care de obicei necesită curenti mult mai mari pentru a funcționa corect. Pentru a gestiona această problemă și pentru a proteja microcontrolerul de orice suprasolicitare, se utilizează drivere de motoare. Driverul de motor preia curentul de la o sursă de alimentare externă care pentru a suporta cerințele de putere ale motoarelor, protejând microcontrolerul și asigurând operarea eficientă a sistemului.

Partea principala a codului:

motor_forward:

```
analogWrite(motorL1, speedL);  
analogWrite(motorL2, 0);  
analogWrite(motorR1, speedR);  
analogWrite(motorR2, 0);
```

motor_backward:

```
analogWrite(motorL1, 0);  
analogWrite(motorL2, speedL);  
analogWrite(motorR1, 0);  
analogWrite(motorR2, speedR);
```

motor_left:

```
analogWrite(motorL1, 0);
analogWrite(motorL2, 0);
analogWrite(motorR1, steerSpeed);
analogWrite(motorR2, 0);
```

motor_right:

```
analogWrite(motorL1, steerSpeed);
analogWrite(motorL2, 0);
analogWrite(motorR1, 0);
analogWrite(motorR2, 0);
```

LOOP:

```
switch (stareCurenta) {
  case OPRIT:
    if(sonar.ping_cm() <= 15 && sonar.ping_cm() >= 5)
    {
      stareCurenta = SUNET_ACTIV;
    }
    else
    {
      stareCurenta = MOTOR_ACTIV;
    }
    break;
  case MOTOR_ACTIV:
    L = digitalRead(pinL);
    C = digitalRead(pinC);
    R = digitalRead(pinR);
    if(C == 0 && L == 0 && R == 0){
      motor_forward();
    }
    else if(C == 0 && L == 1 && R == 1){
      motor_forward();
    }
    else if (C == 0 && L == 1 && R == 0) {
      motor_right();
    }
    else if (C == 1 && L == 1 && R == 0) {
      motor_right();
    }
    else if (C == 1 && L == 0 && R == 1) {
      motor_left();
    }
    else if (C == 0 && L == 0 && R == 1) {
      motor_left();
    }
    else {
      motor_backward();
    }
  }
}
```

```
}  
if(sonar.ping_cm() <= 15 && sonar.ping_cm() >= 5)  
{  
    stareCurenta = OPRIT;  
}  
break;  
case SUNET_ACTIV:  
    motor_stop();  
    if(sonar.ping_cm() > 15 || sonar.ping_cm() < 5)  
    {  
        stareCurenta = OPRIT;  
        break;  
    }  
    else{  
        startPlayback(sounddata_data, sizeof(sounddata_data));  
        delay(4000);  
        stopPlayback();  
    }  
    break;  
}
```

Rezultate Obținute

M-am gandit ca ar fi mai frumos ca in loc sa scoata un sunet repetitiv, robotul chiar sa spuna ceva atunci cand intalneste un obstacol, asa ca acum, in loc de claxon, rosteste fraza "Remove the obstacle".

Link videoclip nou cu robotul alimentat cu ajutorul bateriilor:

<https://youtube.com/shorts/fwjvOJwzmYM?si=7PTZ5CNr6F-3epgj>

Link videoclip mai vechi:

<https://www.youtube.com/shorts/mQUyBDlumzs>

Concluzii

Dificultatile principale la mine au venit in partea de hardware, mai precis in cazul alimentarii pe baza de baterii. Problema a fost ca am achizitionat un modul liniar ridicator de tensiune, care promitea ca poate mari tensiunea data de baterii pana la 28 de V (eu aveam nevoie doar de 9), insa cand am testat cu un voltmetru acesta se oprea pe la 5V si putin orice i-as fi facut. Asa ca, dupa multe ore de munca in care am incercat sa repar asta si nu a mers, am ajuns sa mai adaug o alta baterie corespunzatoare de 9V pentru a alimenta driver-ul de motor.

Download

[emadinu333cc.zip](#)

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

<https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/#:~:text=The%20L298N%20is%20a%20dual,and%20explain%20how%20it%20works.>

https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

<https://koodinkutoja.com/playing-audio-clips-with-arduino/>

<https://www.robotpark.com/image/data/PRO/91470/tcrt5000.pdf>

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2024/aungureanu/ema_carmen.dinu



Last update: **2024/05/26 01:04**