



Arduino Vacuum Cleaner

Proiectare cu microprocesoare

Autor: Timbuc Iulia-Maria

Grupa: 30237

FACULTATEA DE AUTOMATICA
SI CALCULATOARE

11 Ianuarie 2026

Cuprins

1 Descrierea proiectului	2
2 Componente utilizate	2
2.1 Microcontrolere si module de comunicatie	2
2.1.1 Arduino Mega 2560	2
2.1.2 ESP8266 NodeMCU v3	2
2.2 Drivere si motoare	2
2.2.1 Drivere L298N (x2)	2
2.2.2 Motoare DC (x5)	3
2.3 Sistem de aspirare	3
2.4 Alimentare	3
2.4.1 LM2596 - Modul coborator de tensiune	3
2.5 Materiale pentru conexiuni	4
3 Functionalitati si implementare	4
3.1 Software ESP8266	4
3.2 Software Arduino Mega	4
3.3 Fluxul complet al unei comenzi	5
4 Schema electrica si conexiuni	5
5 Concluzii	7
6 Referinte	8

1 Descrierea proiectului

Proiectul este un robot controlat prin WiFi care se poate deplasa in patru directii (inainte, inapoi, stanga, dreapta) folosind comenzi de pe telefon sau laptop. Am folosit un Arduino Mega 2560 pentru controlul motoarelor si un modul ESP8266 pentru conexiunea WiFi.

Ideea de baza este simpla: robotul creeaza propria retea WiFi la care utilizatorul se poate conecta direct cu telefonul. Dupa ce s-a realizat conexiunea, se deschide un browser si se controleaza robotul printr-o pagina web cu butoane, nu trebuie instalata nicio aplicatie.

Sistemul functioneaza in felul urmator: ESP8266-ul se ocupa de partea de WiFi si afiseaza pagina web cu butoanele de control, iar Arduino primeste comenzi si controleaza efectiv cele patru motoare prin doua drivere L298N. Comunicarea intre ESP8266 si Arduino se face printr-un cablu serial simplu, ceea ce asigura ca comenzi ajung rapid la motoare.

2 Componente utilizate

2.1 Microcontrolere si module de comunicatie

2.1.1 Arduino Mega 2560

Arduino Mega se ocupa de:

- Primirea comenziilor de la ESP8266 prin comunicatie seriala
- Controlul celor patru motoare pentru deplasare
- Controlul motorului de aspirare
- Gestionarea vitezei motoarelor prin PWM

Specificatii importante:

- Microcontroller: ATmega2560
- Tensiune: 5V
- Are 54 de pini digitali, dintre care 14 pot face PWM
- 4 porturi Serial
- Memorie: 256 KB

2.1.2 ESP8266 NodeMCU v3

Modulul WiFi care face legatura intre telefon si robot:

- Creeaza retea WiFi proprie (Access Point)
- Ruleaza un server web simplu cu butoanele de control
- Trimit comenzi primite de la telefon catre Arduino prin Serial

Specificatii:

- Cipul ESP8266 integrat
- WiFi 2.4 GHz
- Functioneaza la 3.3V
- Are 4MB memorie Flash

2.2 Drivere si motoare

2.2.1 Drivere L298N (x2)

Se folosesc doua drivere L298N pentru ca fiecare poate controla doar 2 motoare:

- **L298N 1:** controleaza motoarele din fata
- **L298N 2:** controleaza motoarele din spate

Caracteristici L298N:

- Suporta 5V-35V alimentare
- Curent maxim: 2A per motor
- 4 pini pentru directie (IN1-IN4)
- 2 pini PWM (ENA, ENB)
- 2 iesiri per driver (OUT1/OUT2 si OUT3/OUT4)

2.2.2 Motoare DC (x5)

Folosec in total 5 motoare DC:

- **4 motoare pentru deplasare:**
 - Motor fata dreapta
 - Motor fata stanga
 - Motor spate dreapta
 - Motor spate stanga
- **1 motor pentru aspirare:** conectat la o elice montata intr-o sticla care creeaza flux de aer si trage mizeria

2.3 Sistem de aspirare

Am improvizat sistemul de aspirare folosind:

- Un motor DC separat
- O elice mica
- O sticla PET taiata care serveste ca recipient pentru aer

Ideea este ca motorul invarte elica in interiorul sticlei, creand un flux de aer care aspira mizeria de pe jos. Nu este la fel de puternic ca un aspirator profesional, dar functioneaza pentru mizerii usoare.

2.4 Alimentare

Pentru alimentare se foloseste:

- **Baterii:** 2 baterii de 9V conectate pentru motoare
- **Distributie:**
 - Baterie 9V - L298N 1 (motoare fata)
 - Baterie 9V - L298N 2 (motoare spate)
 - Arduino Mega - alimentat prin USB in timpul testarii sau prin baterie separata
 - ESP8266 - alimentat de la pinul 3.3V al Arduino

2.4.1 LM2596 - Modul coborator de tensiune

Am folosit modulul LM2596 pentru a obtine tensiuni stabile din baterie:

- Converteste tensiunea de la baterii (9V) la tensiuni mai mici necesare
- Poate furniza pana la 3A curent
- Are potentiometru pentru ajustarea tensiunii de iesire
- Eficienta ridicata

L-am folosit pentru a alimenta Arduino si ESP8266 la tensiunile corecte fara sa disipez prea multa energie sub forma de caldura.

2.5 Materiale pentru conexiuni

- Fire jumper Dupont Male-Male
- Fire jumper Dupont Male-Female
- Cablu USB pentru programarea Arduino si ESP8266

3 Functionalitati si implementare

Sistemul este implementat pe doua platforme distincte, fiecare cu responsabilitati clare:

3.1 Software ESP8266

Codul pentru ESP8266 este structurat in doua fisiere principale:

esp_code.ino Fisierul principal care implementeaza:

- Configurarea Access Point WiFi (SSID: "Aspirator Iulia", parola: "12345678")
- Initializarea serverului web HTTP pe portul 80
- Definirea rutelor pentru comenzi:
 - / - serveste pagina HTML principală
 - /fata - trimite comanda 'F' (inainte)
 - /spate - trimite comanda 'B' (inapoi)
 - /stanga - trimite comanda 'L' (stanga)
 - /dreapta - trimite comanda 'R' (dreapta)
 - /stop - trimite comanda 'S' (oprire)
- Comunicare seriala la 9600 baud pentru trimiterea comenzilor catre Arduino

webpage.h Fisier header care contine interfata web (HTML/CSS/JavaScript):

- **HTML**: structura paginii cu butoane de control
- **CSS**: design minimalist, butoane centrate
- **JavaScript**: functia `sendCommand()` pentru trimiterea comenzilor

3.2 Software Arduino Mega

Codul Arduino este organizat modular:

Configurare pini

```
// L298N 1 (fata)
IN1_1 = 22, IN2_1 = 23 // Motor fata dreapta
IN3_1 = 24, IN4_1 = 25 // Motor fata stanga
ENA1 = 4, ENB1 = 5      // Enable (PWM)

// L298N 2 (spate)
IN1_2 = 26, IN2_2 = 27 // Motor spate stanga
IN3_2 = 28, IN4_2 = 29 // Motor spate dreapta
ENA2 = 2, ENB2 = 3      // Enable (PWM)
```

Functii principale setup()

- Initializeaza Serial (9600 baud pentru debug)
- Initializeaza Serial1 (9600 baud pentru comunicare cu ESP8266)
- Configureaza toti pinii ca OUTPUT
- Seteaza viteza motoarelor prin PWM (variabila globala: 150/255)
- Opreste toate motoarele la pornire

loop()

- Verifica daca exista date pe Serial1
- Citeste caracterul primit
- Filtreaza caracterele nedorite (newline, carriage return, spatii)
- Executa comanda corespunzatoare printr-o structura switch-case
- Afiseaza comanda primita in Serial Monitor pentru debugging

Functii de miscare

Fiecare functie controleaza cei 8 pini de directie pentru a coordona cele 4 motoare:

- `moveForward()`: toate motoarele in sens inainte
- `moveBackward()`: toate motoarele in sens invers
- `moveLeft()`: motoare dreapta inainte, stanga inapoi (rotire pe loc)
- `moveRight()`: motoare stanga inainte, dreapta inapoi (rotire pe loc)
- `stopMotors()`: toate pinii la LOW (oprire completa)

3.3 Fluxul complet al unei comenzi

1. Utilizatorul apasa butonul "INAINTE" in interfata web
2. JavaScript executa: `fetch('/fata')`
3. ESP8266 primeste cererea HTTP GET la ruta `/fata`
4. ESP8266 executa: `Serial.println("F")`
5. Caracterul 'F' este transmis prin TX (ESP8266) - RX1 pin 19 (Arduino)
6. Arduino detecteaza: `if (Serial1.available())`
7. Arduino citeste: `char comanda = Serial1.read()`
8. Comanda 'F' trece filtrul (nu este newline/space)
9. Switch-case detecteaza: `case 'F'`
10. Se executa: `moveForward()`
11. Pinii IN sunt setati corespunzator pentru fiecare motor
12. Driverele L298N activeaza motoarele
13. Robotul se deplaseaza inainte

4 Schema electrica si conexiuni

In Figura 1 se gaseste schema electrica completa a robotului. Se pot observa conexiunile dintre Arduino Mega , cele doua drivere L298N, ESP8266 , motoarele de deplasare si sursele de alimentare. Toate componente sunt conectate la o masa comună (GND) pentru functionarea corecta a sistemului.

In Figura 2 este prezentata schema simpla pentru motorul de aspirare. Motorul este alimentat direct de la baterie printr-un switch, fiind independent de restul sistemului de control.

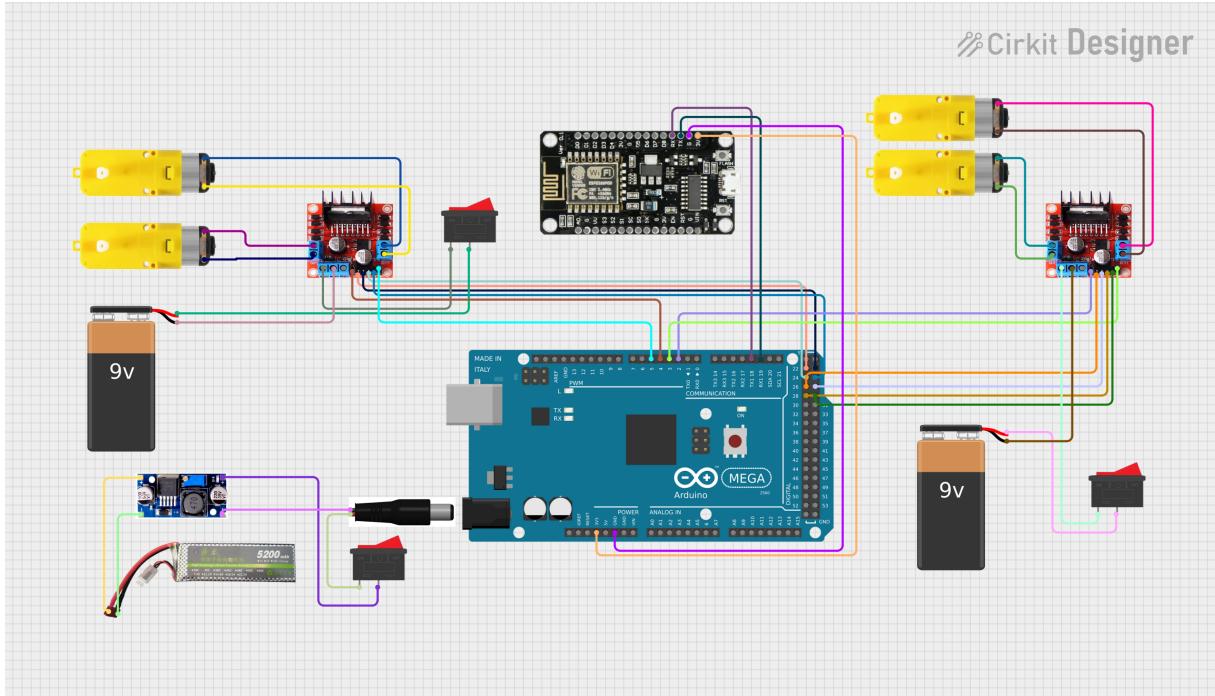


Figura 1: Schema electrica completa a robotului

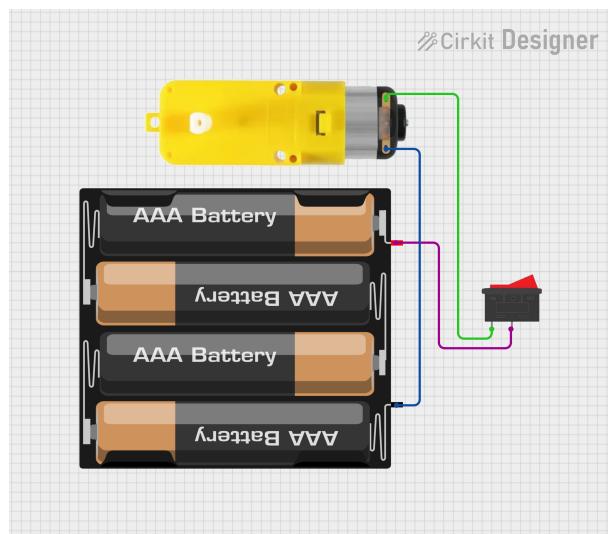


Figura 2: Schema pentru motorul de aspirare

5 Concluzii

Robotul functioneaza conform asteptarilor si se poate controla fara probleme prin WiFi de pe telefon. Interfata web este simpla si rapida - cand apas un buton, robotul raspunde aproape instantaneu.

Configuratia cu rotire pe loc functioneaza bine si permite robotului sa navigheze usor prin spatii inguste. Faptul ca ESP8266 creeaza propria retea WiFi este foarte util - nu trebuie sa depind de un router in apropiere.

Pe parcurs am intampinat cateva probleme:

- Arduino primea caractere in plus dupa fiecare comanda (newline si carriage return), ceea ce oprea motoarele in mod nedorit. Am rezolvat adaugand un filtru in cod care ignora aceste caractere.
- Un motor (fata dreapta) mergea in directia opusa pentru ca era conectat invers. Am inversat firele la OUT1 cu OUT2 si problema s-a rezolvat.
- Initial ESP8266 nu era detectat de Arduino IDE pentru ca imi lipsea driverul CH340. Dupa ce l-am instalat, totul a mers ok.

Proiectul mi-a aratat ca se poate face un robot functional cu componente relativ ieftine si ca separarea clara intre partea de WiFi (ESP8266) si controlul motoarelor (Arduino) face sistemul mai usor de inteles si de modificat.

Am invatat multe despre comunicarea seriala, controlul motoarelor prin PWM si cum sa creez o interfata web simpla. Robotul poate fi imbunatatit cu multe functii noi si poate servi ca baza pentru proiecte mai complexe.

6 Referinte

1. Tutorial control robot prin WiFi
<https://www.youtube.com/watch?v=UU7VVbEZulw>
2. Tutorial de unde am luat partea de aspirare
<https://www.youtube.com/watch?v=hHIIWpZcYNs&t=162s>
3. Documentatie Arduino Mega 2560
<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
4. Conectare ESP8266 la WiFi
<https://gilkut.com/nodemcu/how-to-connect-esp8266-to-wi-fi-step-by-step-guide-for-beginners>
5. Conectare L298N la Arduino
<https://www.hackster.io/ryanchan/how-to-use-the-l298n-motor-driver-b124c5>