

# Masina de tuns iarba

## Introducere

Proiectul constă într-o mașină de jucărie controlată prin Bluetooth, de pe telefonul mobil, echipată cu LED-uri pentru a imita farurile și stopurile și cu o elice atașată pentru tăiatul ierbii. În plus față de aceste caracteristici, am adăugat și un LCD pentru a afișa viteza la care sunt controlate motoarele.

Scopul este de a oferi o modalitate distractivă și practică pentru îndeplinirea unei sarcini, precum tăierea ierbii în grădină.

## Descriere generală

Modulul HC-05 primește comenzile Bluetooth de la aplicație și le transmite la microcontroller folosind interfața USART. Microcontrollerul interpretează aceste comenzi și va trimite mai departe către driver-ul motoarelor (L298N) pentru a indica direcția și viteza motoarelor, folosind interfața PWM. LCD-ul va afișa viteza motoarelor, aceasta fiind selectată din aplicație.



## Hardware Design

Piese utilizate:

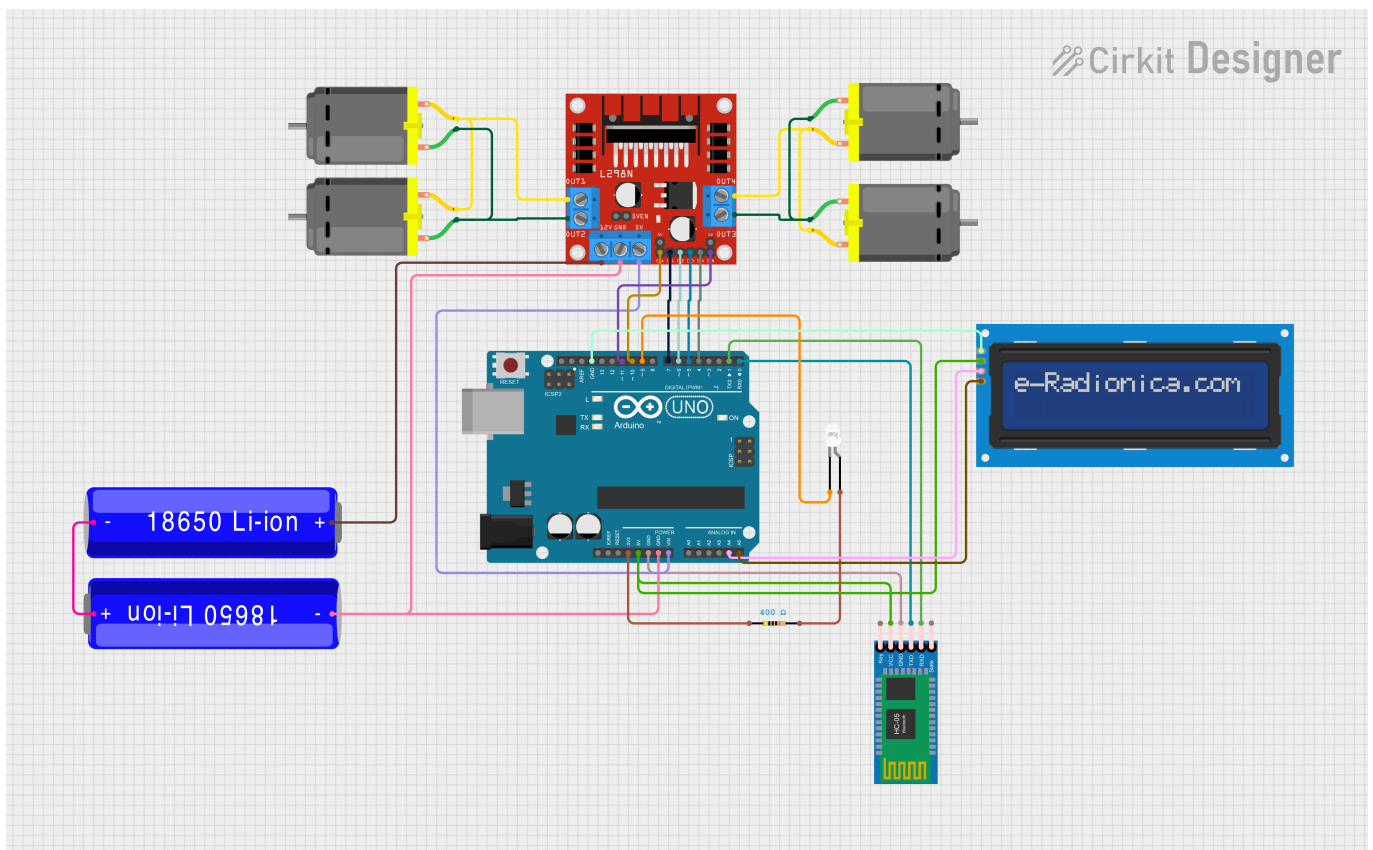
- Arduino UNO (ATmega328p)
- Modul bluetooth HC-05
- Modul cu Driver de Motoare Dual L298N
- Sasiu de masina cu 4 motoare (direcție și deplasare)
- 4 led-uri
- Cabluri cu Conectori Tata-Tata, Tata-Mama
- 16x2 LCD screen cu interfața I2C
- Rezistență de 400 Ohmi
- Acumulator

Piese pentru lama de tăiat iarba:

- Baterie 9v

- Motor DC
- Elice de plastic
- Fire
  
- ATmega328P: Este responsabil pentru interpretarea comenzilor Bluetooth primite de la modulul HC-05 și pentru controlul funcțiilor mașinii și a LED-urilor.
- Modulul Bluetooth HC-05: Este folosit pentru a permite comunicarea fără fir între telefonul mobil și mașina. Transmite comenzile primite de la telefon către ATmega328P.
- Modulul cu Driver de Motoare Dual L298N: Folosit pentru a controla motoarele care conduc direcția și deplasarea mașinii
- 4 LED-uri: Acestea sunt utilizate pentru a imita farurile și stopurile mașinii.
- Cabluri cu Conectori Tata-Tata, Tata-Mama: Utilizate pentru a conecta componentele între ele în cadrul circuitului.
- 16x2 LCD screen cu interfata I2C: Folosit pentru a afișa viteza motoarelor.
- Acumulator: Fornizează energia necesară pentru funcționarea mașinii și a componentelor electronice asociate.

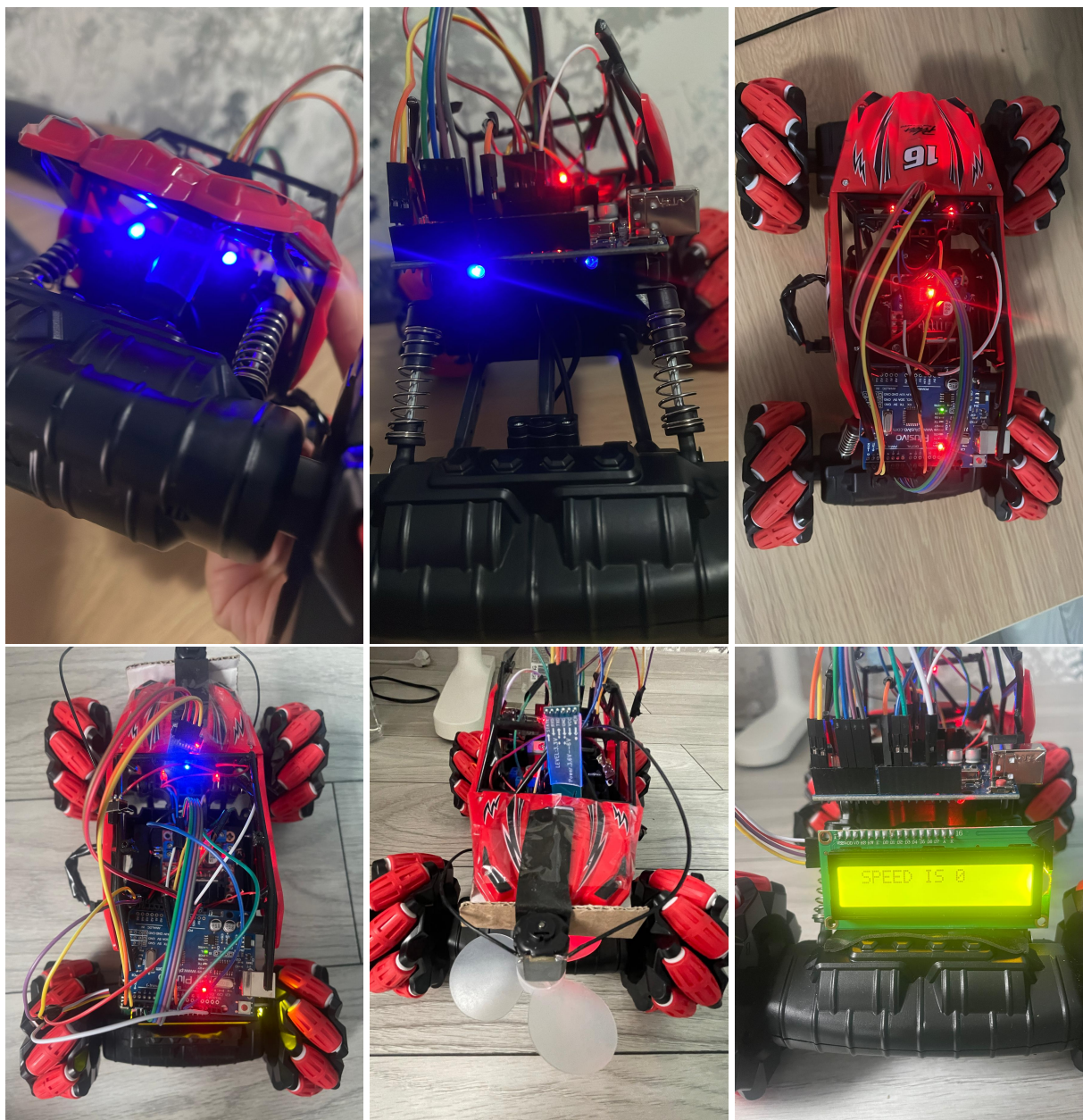
## Schema electrica



# Schema electrica elice



## Componentele conectate



## Pini folositi

\* Pentru a folosi USART in comunicarea dintre microcontrolerul ATmega328P și modulul Bluetooth HC-05,conectez piniul RX de la modul la pinul TX de la microcontroler, si TX la RX.De asemenea, VCC ul de la modul il conectez la pinul de 5V de la microcontroler pentru alimentare si GND ul la GND.

- \* Motoarele sunt conectate la modulul L298N prin pinii out1,out2,out3 si out4 pentru a controla directia lor.
- \* Pentru a functiona tot ansamblul, cele 2 acumuloare sunt conectate la pinul de 12V de la modulul L298N, care ofera si alimentare microcontrolerului prin pinul de 5V.
- \* Comenzile primite de la modulul HC-05 sunt trimise la modulul care controleaza motoarele prin pinii de semnale digitale Am ales pinii 4,5,6,7 de la microcontroler
- \* Pinii 10 si 11 de la microcontroler conectati la pinii ENA si ENB de la modulul L298N controleaza viteza motoarelor prin PWM
- \* LED urile sunt conectate la pinul digital 9 de la microcontroler si alimentarea la pinul de 3.3V
- \* Pinii VCC SI GND de la LCD cu interfata I2C sunt conectati la pinii de 5v si, respectiv GND ai microcontrolerului.
- \* Pinii SDA SI SCL ai LCD ului sunt conectati la microcontroler prin pinii A4 si A5(SDA SI SCL)

## Software Design

- **Mediu de dezvoltare:** Arduino IDE
- **biblioteci folosite:** Wire.h si LiquidCrystal\_I2C.h (pentru controlul motoarelor, comunicatie I2C și afisarea pe LCD.)
- **Pentru viteza motoarelor** am folosit o variabila globala.
- Am definit pinii pentru controlul motoarelor(**in1,2,3,4**), pentru viteza(**ena, enb (PWM)**) si pentru led uri

**Funcția setup** este utilizată pentru a inițializa hardware-ul și setările necesare.

```
void setup() {
```

```
Serial.begin(9600);  
DDRD |= (1 << in1) | (1 << in2) | (1 << in3) | (1 << in4);  
DDRB |= (1 << ena) | (1 << enb) | (1 << led);  
PORTB |= (1 << led);  
analogWrite(ena, 0);  
analogWrite(enb, 0);
```

```
lcd.init();  
lcd.backlight();  
lcd.setCursor(1, 0);  
lcd.print("SPEED IS 0");
```

```
}
```

- Inițializare comunicarea serială (Serial.begin(9600))

- **in1, in2, in3, in4**: pini digitali setați ca ieșiri, pentru a controla direcția motoarelor.
- **ena, enb**: pini de ieșire care controlează viteza motoarelor folosind PWM.
- **led**: pin de ieșire pentru LEDuri, pentru semnalizarea diferitelor stări.
- **analogWrite**: setează viteza inițială a motoarelor la 0, oprind motoarele la pornirea sistemului.

**Funcția loop** este responsabilă pentru citirea comenzilor de la interfața serială și pentru controlarea comportamentului vehiculului pe baza acestor comenzi. Se verifică dacă există date disponibile pe interfața serială. Dacă da, se citește o valoare care reprezintă ce funcție trebuie apelată. Se setează viteza motoarelor și se afișează pe LCD valoarea ei.

```
void loop() {
```

```
  if (Serial.available() > 0) {
    char value = Serial.read();
    analogWrite(ena, Speed);
    analogWrite(enb, Speed);
    lcd.clear();
    lcd.print("SPEED IS ");
    lcd.print(Speed);
    lcd.print("  ");

    if (value == 'B') {
      Forward();
    }
    else if (value == 'F') {
      Back();
    }
    else if (value == 'L') {
      Left();
    }
    else if (value == 'R') {
      Right();
    }
    else if (value == 'X') {
      Speed = 255;
    }
    else if (value == 'Y') {
      Speed = 120;
    }
    else if (value == 'S') {
      Stop();
    }
    else if (value == 'M' || value == 'N')
    {
      PORTB &= ~(1 << led);
    }
    else if (value == 'm' || value == 'n')
    {
      PORTB |= (1 << led);
    }
  }
```

```
}
```

```
}
```

Funcțiile **“Forward()”**, **“BACK()”**, **“LEFT()”**, **“RIGHT()”** și **“STOP()”** sunt responsabile de controlul mișcării mașinii.

```
void Forward() {
```

```
PORTD |= (1 << in1) | (1 << in3);  
PORTD &= ~((1 << in2) | (1 << in4));
```

```
}
```

```
void Back() {
```

```
PORTD |= (1 << in2) | (1 << in4);  
PORTD &= ~((1 << in1) | (1 << in3));
```

```
}
```

```
void Left() {
```

```
PORTD |= (1 << in1) | (1 << in4);  
PORTD &= ~((1 << in2) | (1 << in3));
```

```
}
```

```
void Right() {
```

```
PORTD |= (1 << in2) | (1 << in3);  
PORTD &= ~((1 << in1) | (1 << in4));
```

```
}
```

```
void Stop() {
```

```
PORTD &= ~((1 << in1) | (1 << in2) | (1 << in3) | (1 << in4));
```

```
}
```

## Rezultate Obținute

[https://drive.google.com/file/d/1ADtn8xu3hT9tyJ2f52uw-6Ngnt1La\\_3Z/view?usp=drive\\_link](https://drive.google.com/file/d/1ADtn8xu3hT9tyJ2f52uw-6Ngnt1La_3Z/view?usp=drive_link)

## Concluzii

Finalizarea acestui proiect a fost o experiență extrem de satisfăcătoare, oferindu-mi ocazia de a integra și aplica cunoștințele de hardware și software într-un mod creativ și util. Proiectarea și construirea unei mașini de jucărie controlate prin Bluetooth, echipată cu LED-uri pentru faruri și stopuri, și cu o elice pentru tăiatul ierbii, mi-au permis să explorez numeroase aspecte ale ingineriei și programării. Afișarea vitezei pe un ecran LCD a adăugat un plus de interactivitate și funcționalitate proiectului. Acest proiect nu doar că mi-a îmbogățit abilitățile tehnice, dar mi-a și stimulat dorința de a lucra la proiecte similare în viitor.

## Download

[proiect\\_pm.zip](#)

## Bibliografie/Resurse

### Resurse Software:

- **Arduino Libraries**
- **LiquidCrystal\_I2C Library Documentation**
- **Wire Library Documentation**

### Resurse Hardware:

- **Datasheet Arduino Uno**
- **Datasheet L298N Motor Driver**
- **Datasheet HC-05**
- <https://www.youtube.com/watch?v=CvqHkXeXN3M>
- <https://www.youtube.com/watch?v=bNZv3dZpmSU>
- [https://www.youtube.com/watch?v=ykgt4v7\\_4y4](https://www.youtube.com/watch?v=ykgt4v7_4y4)
- <https://www.youtube.com/watch?v=E2sTbpFsvXI>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/vstoica/valentin.ionescu02>



Last update: **2024/05/26 19:10**