

# Ioan Teodorescu: Card Shuffler

**Nume: Teodorescu Ioan**

**Grupa: 333CB**

**Indrumator: Daniel Dosaru**

## Introducere

- Ce face?

Proiectul meu constă în implementarea unui dispozitiv automat care amestecă un pachet de cărți de joc.

- Care este scopul lui?

Scopul este de a oferi o soluție convenabilă și rapidă pentru amestecarea cărților, eliminând nevoia de a le amesteca manual. De asemenea, nu mai este posibilă trișarea!

- Care a fost ideea de la care ați pornit?

Am văzut un dispozitiv asemănător, însă respectivul produs era manual și trebuia folosită manivela pentru a amesteca pachetul de cărți. M-am gândit că ar fi interesant să transform acest proces într-unul automat.

- De ce credeți că este util pentru alții și pentru voi?

Cum am menționat mai sus, nu mai este posibilă trișarea. Astfel, putem avea un joc cât mai corect. În plus, aparatul poate fi util pentru jucătorii de poker sau blackjack; sau chiar pentru cazinouri și cluburi de jocuri.

## Descriere generală

La pornirea dispozitivului:

1. Introducerea cărților: Utilizatorul va fi solicitat, prin intermediul ecranului, să introducă cărțile în ambele deschideri de sus, pregătind-le pentru amestecare. Pornirea procesului va fi imposibilă până la introducerea cărților.
2. Senzori de distanță: Senzori de distanță ultrasonic sunt poziționați în ambele zone. Când distanța scade semnificativ (indicând prezența cărților), se va putea porni procesul.
3. Confirmarea: Informația va fi confirmată prin:

- Un mesaj pe ecran

- Aprinderea unui LED

Pornirea amestecării:

1. Comandă prin telefon: Utilizatorul va trimite comanda de pornire a amestecării prin intermediul telefonului său.
2. Activarea motoarelor: Motoarele dispozitivului vor porni și vor amesteca cărțile.
3. Oprire automată: Amestecarea se va opri automat când nu mai sunt cărți rămase într-una dintre deschideri.



## Hardware Design

### Lista piese:

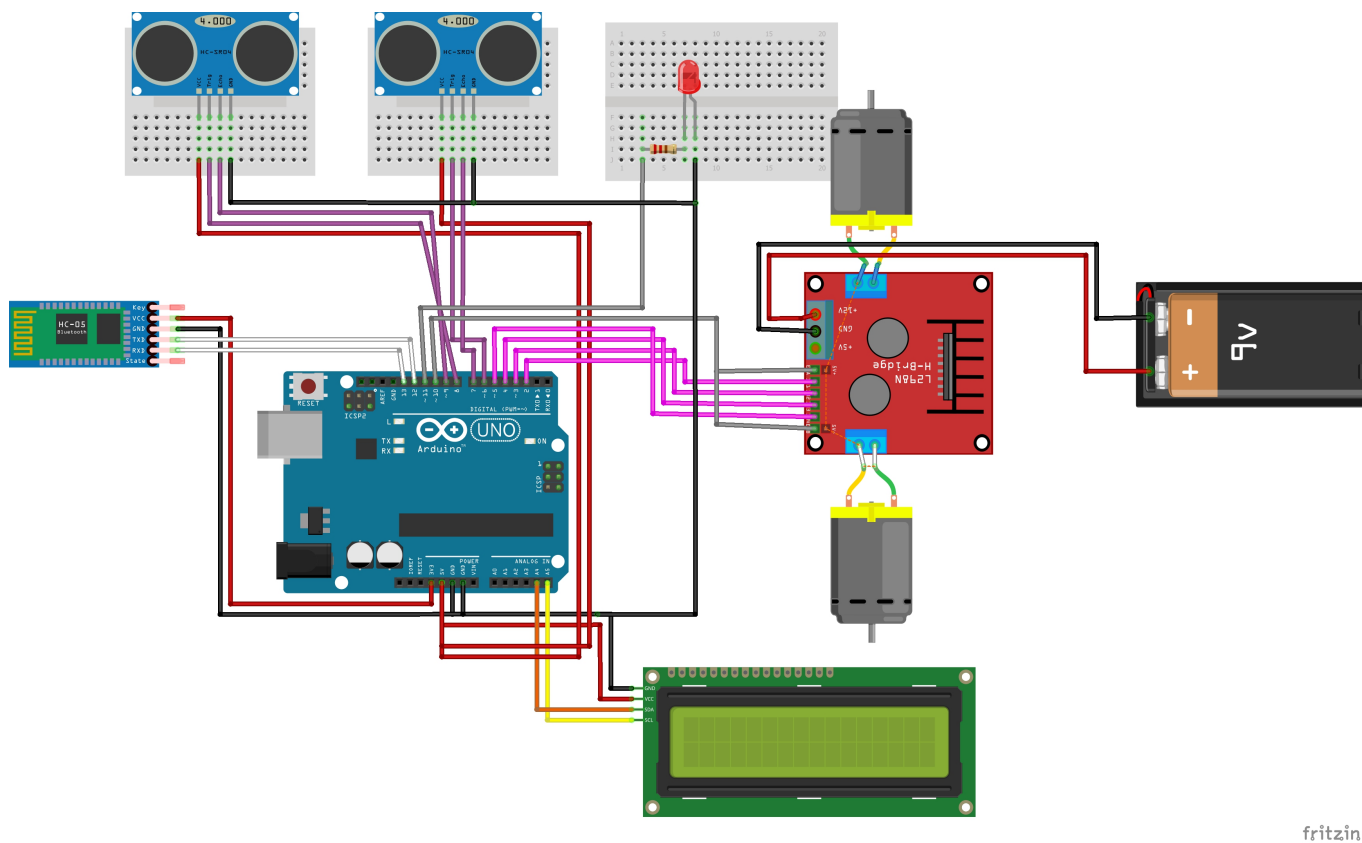
- Arduino Uno R3
- 2 x Motor DC 1 AX
- 2 x Senzor ultrasonic HC-SR04
- Baterie 9V
- Rezistenta 220  $\Omega$  + LED
- Modul Bluetooth 4.0 BLE CC2540 CC2541
- Ecran LCD 1602 + Adaptor I2C pentru LCD 1602
- L298N Dual Motor Driver

### Conectivitate

- P2: Se conecteaza la portul IN1 al driverului L298N
- P3: Se conecteaza la portul IN2 al driverului L298N
- P4: Se conecteaza la portul IN3 al driverului L298N
- P5: Se conecteaza la portul IN4 al driverului L298N
- P6: Se conecteaza la portul TRIG al primului senzor HC-SR04
- P7: Se conecteaza la portul ECHO al primului senzor HC-SR04
- P8: Se conecteaza la portul TRIG al celui de-al doilea senzor HC-SR04
- P9: Se conecteaza la portul ECHO al celui de-al doilea senzor HC-SR04
- P10: Se conecteaza la porturile ENA si ENB din driverul L298N
- P11: Se conecteaza la breadboard pentru aprinderea LED-ului
- P12: Se conecteaza la portul TXD al modulului Bluetooth
- P13: Se conecteaza la portul RXD al modulului Bluetooth
- A4: Se conecteaza la portul SCL al modulului I2C
- A5: Se conecteaza la portul SCL al modulului I2C

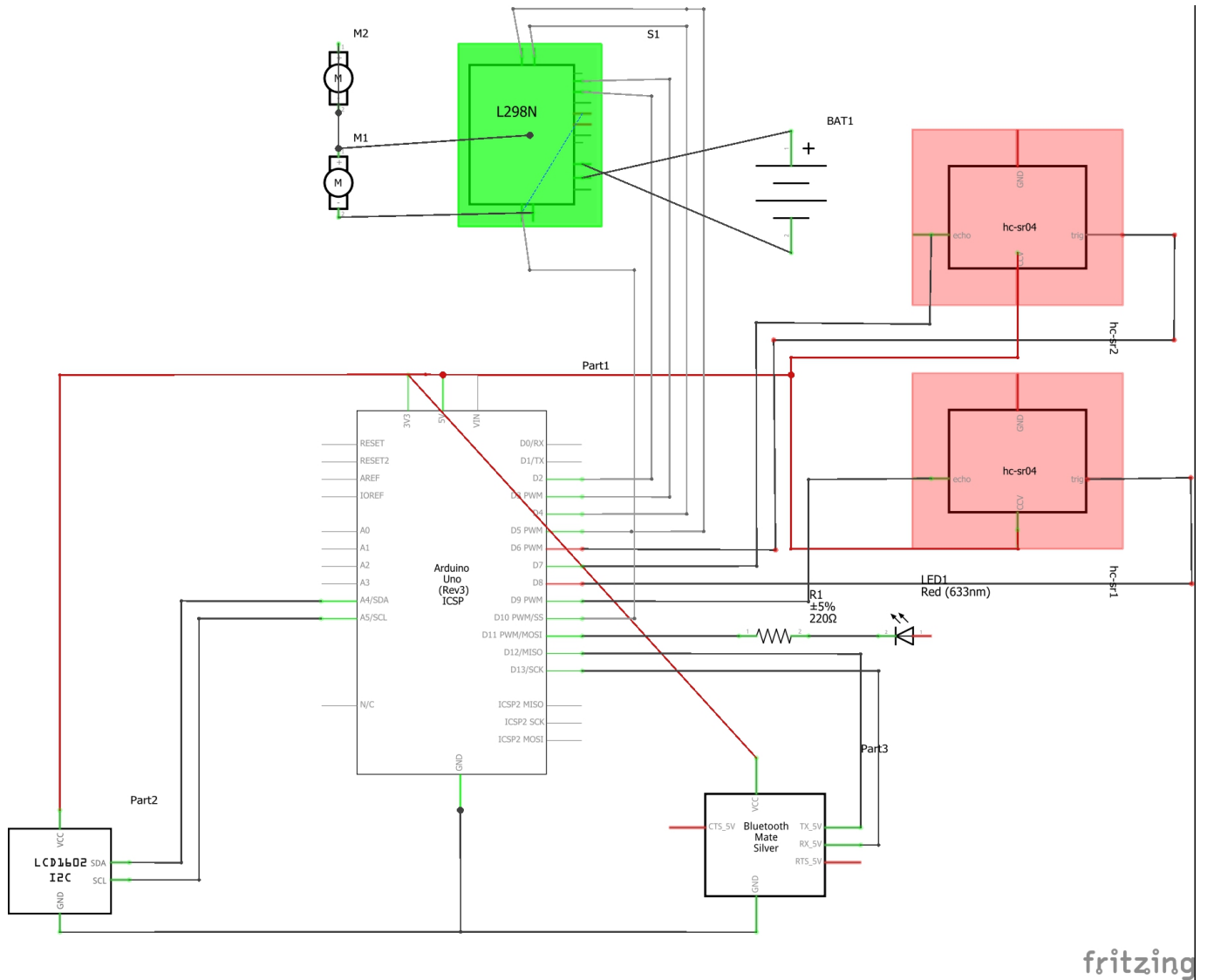
# Schema Electrica

Arduino Uno ofera PWM (Pulse Width Modulation) pe pinurile 3, 5, 6, 9, 10 si 11. Astfel, pentru a controla viteza celor doua motoare, am folosit pinul 10, care, adaugat in breadboard am scos alte 2 fire pentru a le conecta pe porturile ENA si ENB din L298N. Modulii Bluetooth funcționează la o tensiune de alimentare de 3.3V, contribuind la menținerea unui consum redus de energie.

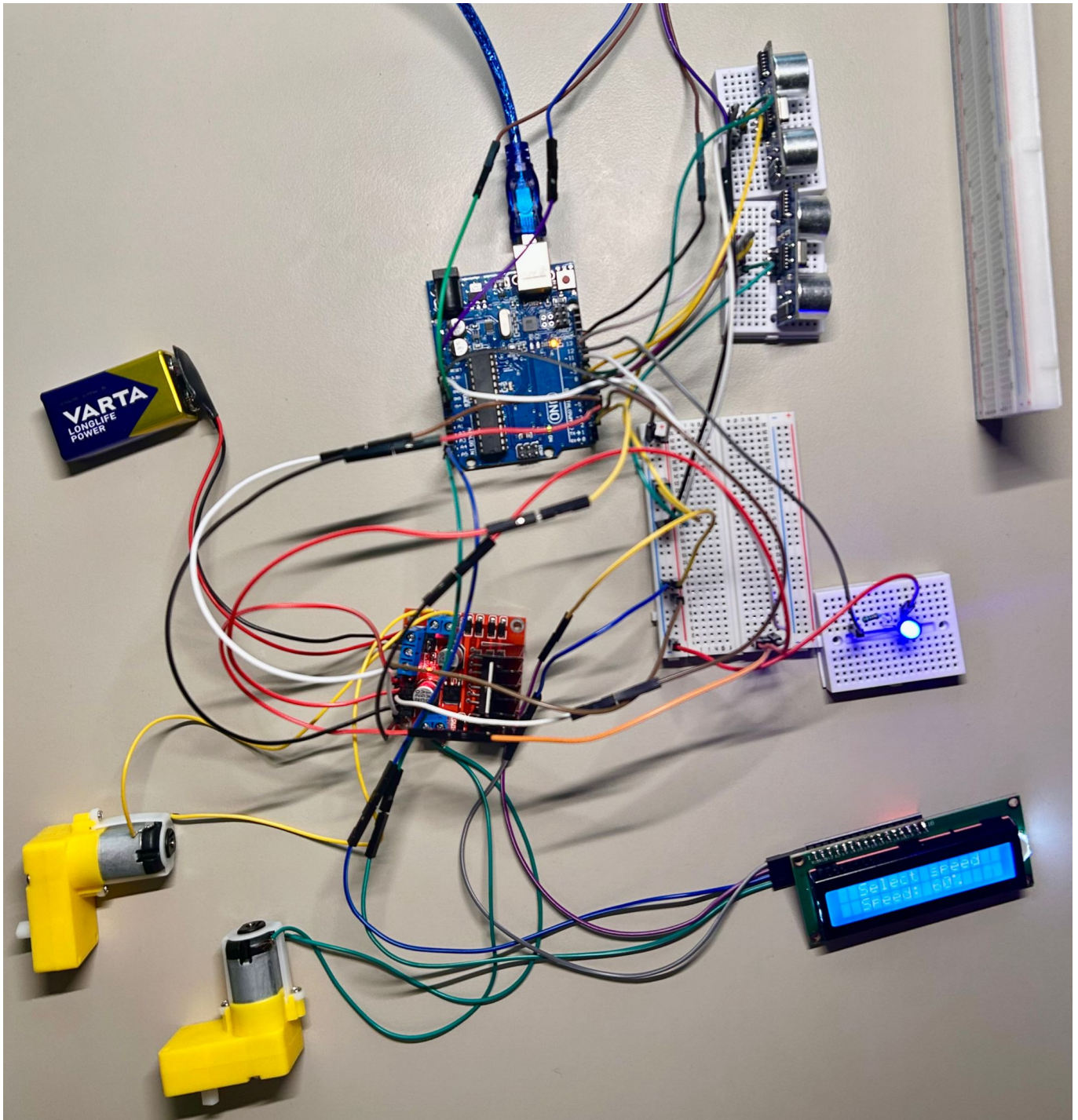


fritzing

# Diagrama Electrica



## Implementare schema



## Software Design

Pentru implementarea software am folosit Arduino IDE. Pe parcursul implementarii proiectul, m-am decis să ofer disponibilitatea utilizatorilor de a gestiona viteze de mestecare a cartilor.

## Biblioteci alese

- LiquidCrystal\_I2C
  - Această bibliotecă este utilizată pentru a controla un afișaj LCD prin protocolul I2C, economisind

pinii de pe microcontroller și simplificând conexiunile hardware.

- HCSR04
  - Această bibliotecă facilitează interfațarea cu senzorii de distanță ultrasonici HCSR04, permițând măsurarea distanței într-un mod simplu și precis.
- SoftwareSerial
  - Aceasta permite comunicarea serială pe pinii digitali selectați de utilizator, fiind utilă pentru modulele care necesită o interfață serială suplimentară, cum ar fi modulul Bluetooth.

## Laboratoare folosite

- USART
  - UART ajută la interfațarea și comunicarea eficientă cu modulul Bluetooth. Am utilizat biblioteca `SoftwareSerial` pentru a crea o interfață serială suplimentară pe pinii digitali 12 și 13 (Rx și Tx) pentru modulul. Apoi, comunicarea serială a fost configurată la o viteză de baud de 9600 bps, atât pentru seriala hardware, cât și pentru cea software. Aceasta este o valoare comună și fiabilă pentru transferul de date între microcontroller și modulul Bluetooth. Utilizând `bt.read()`, dispozitivul primește date de la modulul Bluetooth, permițându-mi să controleze dispozitivul
- PWM
  - PWM ajută la a varia tensiunea dată motoarelor, în mod controlat. În cod, `enPin` este configurat ca pin de ieșire PWM care controlează viteza motoarelor. Utilizând funcția `analogWrite(enPin, motorSpeed)`, valoarea PWM (cuprinsă între 0 și 255) determină viteza motorului.
- I2C
  - I2C este foarte important în proiectul acesta, având ca principal scop economisirea pinilor microcontroller-ului și simplificarea conexiunilor hardware. Pentru început, am folosit un I2C Scanner pentru a găsi adresa ecranului (în acest caz, 0x27). Cu ajutorul `LiquidCrystal_I2C`, care a facilitat procesul de controlare al ecranului, am reușit să controlez ecranul cu succes.

## Explicarea scheletului proiectului

- `setup()`
  - Se initializează configurația pinilor și initializarea componentelor (Ecran LCD, modul Bluetooth).
- `updateInsertedState()`
  - Verifică distanța față de cel mai apropiat obiect, pentru fiecare senzor. Dacă distanța relativă este mai mică decât 10 la ambii senzori, înseamnă că avem cârți introduse, iar `inserted` e 1, altfel, 0.
- `displaySpeedSelection()`
  - Afisează pe ecran mesajul de selectare a vitezei
- `updateSpeedDisplay()`
  - Pentru funcția de mai sus, actualizează doar procentajul.
- `displayShufflingMessage()`
  - Afisează faptul că se amestecă pachetul.
- `doneMessage()`
  - Confirmă terminarea procesului.
- `startMotors(int motorSpeed)`
  - Porneste motoarele.

- `displayErrorMessage()`
  - Afișează un mesaj de eroare în cazul în care vrem să începem procesul, însă nu am nimic în compartimente.
- `stopMotors()`
  - Oprește motoarele.
- `loop()`
  - În primul rând, calculează cele 2 distanțe și verificăm dacă starea s-a schimbat (e diferită de starea cea veche). În caz afirmativ, vedem care este valoarea lui `inserted` (1 - avem cv în compartiment, 0 - nu e nimic) și se afișează mesajele respective. Apoi, verificăm dacă am primit ceva de la terminal (modul ble); Dacă da, extragem ce am citit. Dacă avem un număr din intervalul [1, 6], vom actualiza viteza motoarelor; dacă primim 'S', vom da start la motoare. În momentul în care distanțele de la ambii senzori cresc, se oprește motorul.

## Rezultate Obținute

Mic demo înainte de a realiza suportul final: <https://www.youtube.com/watch?v=jcgRYc394oc>





Rezultat final: <https://www.youtube.com/watch?v=CSkBXTHyeMI>

Tin să menționez faptul că suportul nu este cel mai perfect, iar procesul de amestecare nu este cel mai bun.

## Download

Arhiva contine:



- Fisierul `card_shuffler.ino` care contine logica proiectului
- `HCSR04-ultrasonic-sensor-lib-2.0.2.zip` arhiva pentru biblioteca HCSR04
- `LiquidCrystal_I2C-1.1.2.zip` arhiva pentru biblioteca LiquidCrystal\_I2C

Arhiva → [proiect\\_pm.zip](#)

## Jurnal

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

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/ddosaru/ioan.teodorescu1206>



Last update: **2024/05/27 12:47**