

Sistem de Monitorizare a Vehiculului

PROGRAMAREA APLICȚIILOR INTERNET

MIHĂILĂ BOGDAN-IULIAN

Table of Contents

CAPITOLUL 1. INTRODUCERE.....	2
CAPITOLUL 2. SOFTWARE.....	3
CAPITOLUL 3. HARDWARE.....	8
CAPITOLUL 4. CONCLUZII.....	19
ANEXE:.....	20



CAPITOLUL 1. INTRODUCERE

Proiectul de față constă în dezvoltarea unei aplicații web interconectate cu o mașină echipată cu o placă Arduino. Scopul principal al acestei aplicații este monitorizarea și gestionarea parametrilor de funcționare ai mașinii în timp real. Placa Arduino, integrată în sistemul mașinii, transmite informații precum rotațiile pe minut (RPM), tensiunea electrică și coordonatele senzoriale (X, Y, Z). Aplicația oferă o platformă intuitivă și sigură pentru afișarea acestor date, asigurând o interfață prietenoasă pentru utilizatori.

Obiectivele proiectului

1. Monitorizarea parametrilor tehnici ai mașinii:

- Citirea și afișarea în timp real a valorilor furnizate de placa Arduino, inclusiv:
 - **RPM (rotații pe minut):** Pentru a urmări viteza motorului.
 - **Tensiunea electrică:** Indicativ al stării sistemului electric.
 - **Coordonatele senzoriale (X, Y, Z):** Pentru a evalua mișcarea sau înclinarea.

2. Sistem de logare securizat:

- Implementarea unei pagini de logare pentru a permite accesul doar utilizatorilor autorizați.
- Prevenirea accesului neautorizat și protecția datelor mașinii.

3. Vizualizare grafică a datelor:

- Crearea de grafice pentru reprezentarea vizuală a variațiilor parametrilor (ex: RPM și tensiune).
- Actualizarea în timp real a graficelor pentru a oferi o imagine completă și dinamică.

4. Gestionarea conexiunii seriale cu placa Arduino:

- Conectarea și deconectarea ușoară a dispozitivului prin intermediul aplicației.
- Citirea datelor trimise prin portul serial și afișarea acestora într-o consolă interactivă.

5. Detectarea și gestionarea erorilor:

- Identificarea problemelor în conexiunea cu placa Arduino și afișarea mesajelor corespunzătoare.
- Crearea unei zone dedicate pentru afișarea erorilor sau mesajelor de succes.

CAPITOLUL 2. SOFTWARE

Frontend (HTML + CSS)

Această parte definește design-ul și interfața grafică cu utilizatorul.

1. Designul formularului (form)

- Formularul este stilizat cu CSS pentru a avea o interfață modernă, incluzând:
 - Efecte de **blur** și transparență (`backdrop-filter: blur(10px);` și `rgba`).
 - Margini rotunjite (`border-radius: 10px;`) și o umbră subtilă (`box-shadow`).
- Culoarea textului și a elementelor input este adaptată unui fundal întunecat, utilizând fontul *Poppins*.

2. Elementele interactive (input, button, și social)

- Câmpurile de input (nume de utilizator și parolă) sunt definite pentru a avea un stil minimalist:
 - Fundal translucid, margini rotunjite și text gri deschis pentru placeholder.
- Butonul (button) este configurat să aibă un fundal alb, cu text vizibil pe fundal întunecat.
- Secțiunea socială (butonul pentru rețele sociale) include un efect de hover pentru o mai bună experiență a utilizatorului.

3. Elemente decorative (forma gradientelor)

- Două forme decorative (`.shape`) sunt plasate în fundal, folosind gradientele definite prin CSS pentru un efect vizual atractiv.

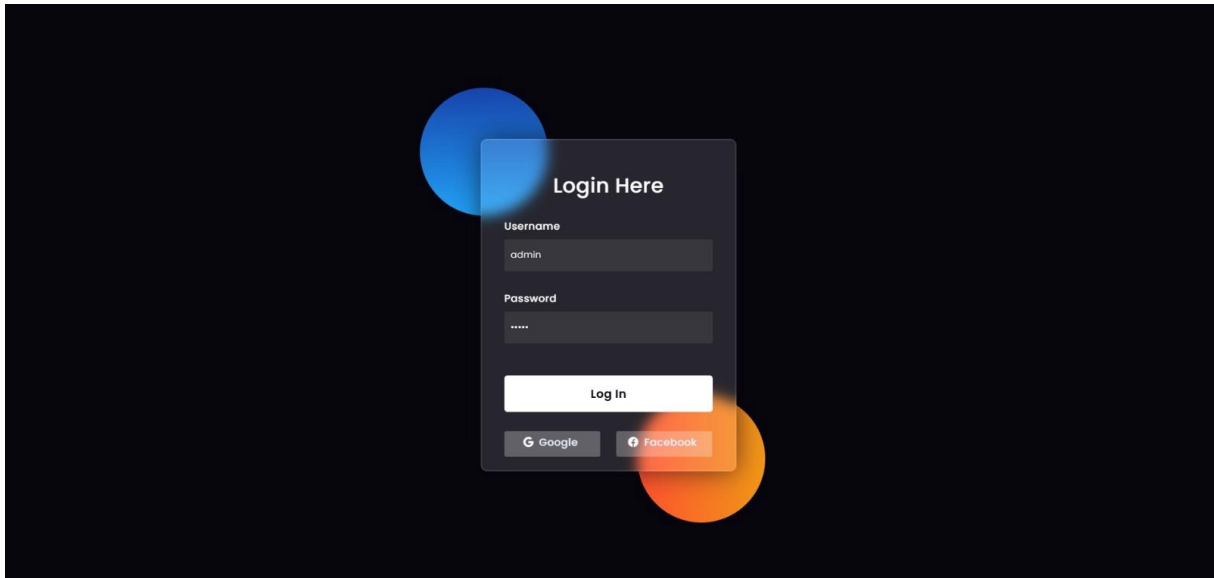
Backend Logic (JavaScript)

Această parte gestionează logica de autentificare. Este o abordare simplă, utilizată mai ales pentru scopuri de demonstrație.

Funcționalitatea principală:

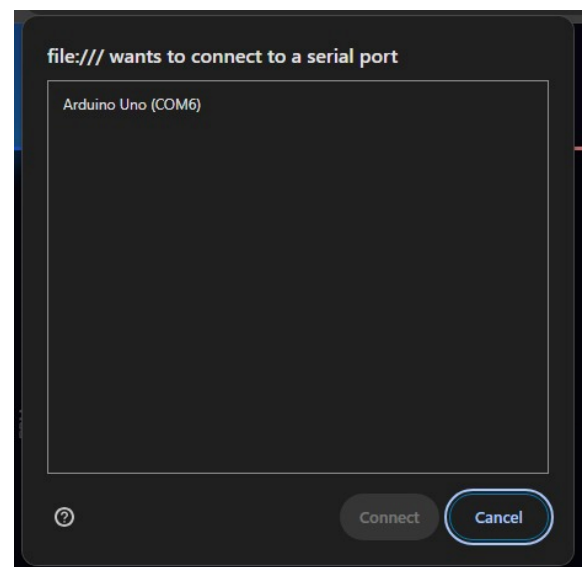
- Evenimentul submit al formularului este prevenit prin `event.preventDefault()` pentru a împiedica trimiterea datelor înainte de validare.
- Logica verifică dacă datele introduse de utilizator corespund cu o combinație predefinită (ex.: username admin și parola admin).
- În funcție de validare:
 - Dacă sunt corecte: Se deschide o pagină goală (`window.open('about:blank', '_self')`).
 - Dacă sunt greșite: Apare un mesaj de eroare (`alert("Username sau parola incorrectă. Încearcă din nou.")`).

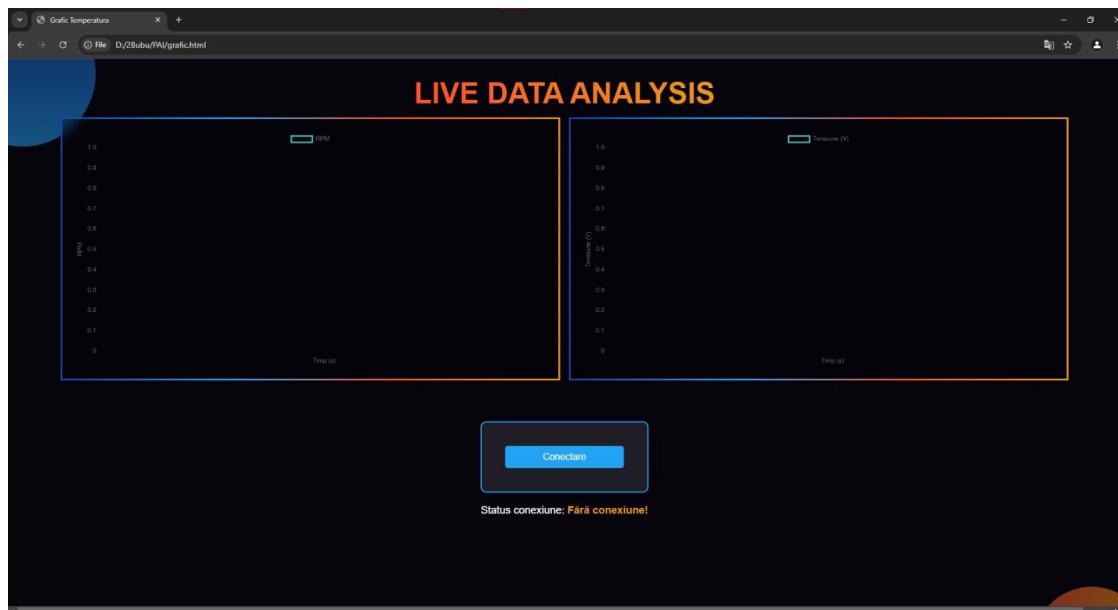
Acest site este o aplicație web care permite **monitorizarea în timp real a datelor primite de la un dispozitiv conectat printr-un port serial**, cum ar fi un **Arduino Uno**. Principalele funcționalități ale aplicației sunt următoarele:



1. Conexiunea la dispozitivul serial

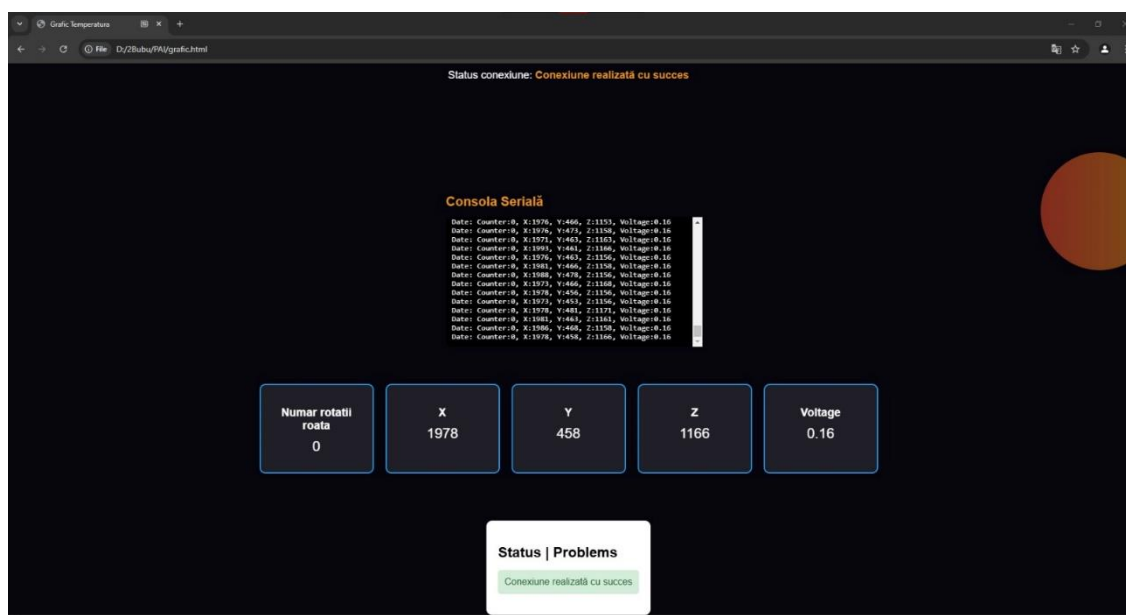
- Utilizatorul poate stabili o conexiune cu un dispozitiv hardware folosind butonul **"Conectează-te"**.
- Dacă conexiunea este realizată cu succes:
 - Se afișează mesajul **"Conexiune realizată cu succes"**.
 - Butonul de conectare dispăre și este înlocuit de un buton **"Deconectare"**.





2. Citirea și procesarea datelor seriale

- După stabilirea conexiunii, aplicația începe să citească **datele transmise de dispozitiv** prin portul serial.
- Datele sunt procesate și afișate în interfața web:
 - **Counter:** Un contor care arată numărul de evenimente sau impulsuri măsurate.
 - **X, Y, Z:** Valori asociate senzorilor sau altor măsurători relevante.
 - **Voltage:** Nivelul de tensiune detectat.
 - Toate aceste date sunt actualizate în timp real într-o secțiune vizuală.



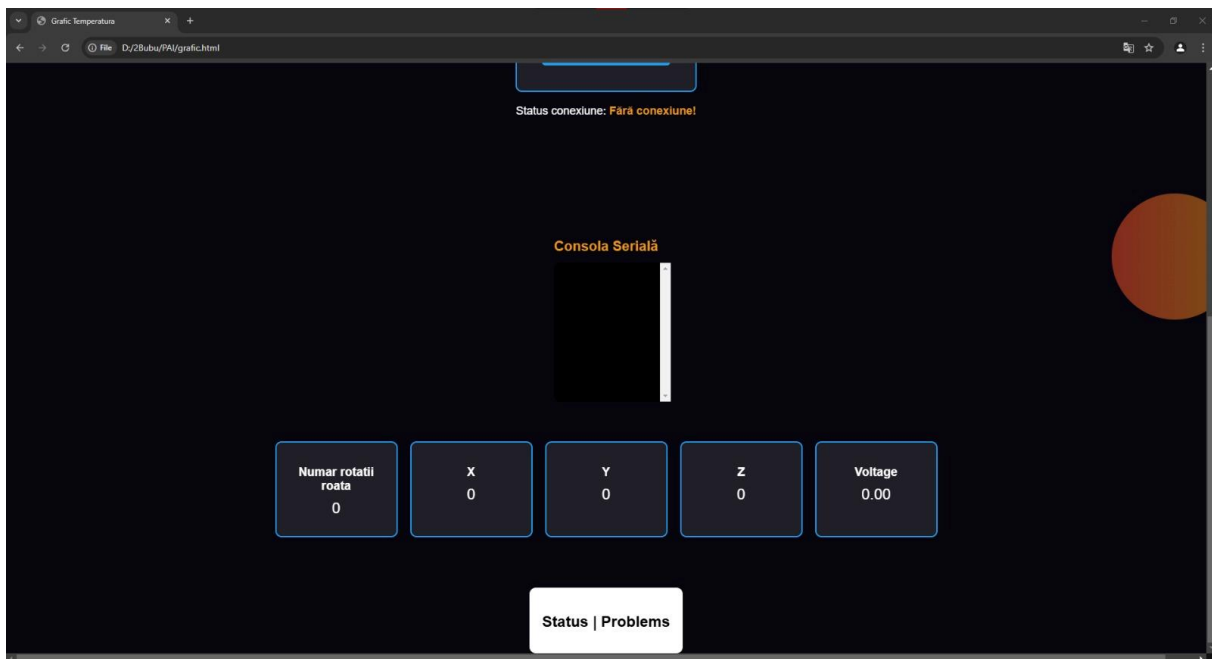
3. Vizualizarea grafică a datelor

- Aplicația afișează două grafice dinamice:
 - **RPM (rotatii pe minut):** Calculat pe baza impulsurilor măsurate și a timpului dintre citiri.
 - **Tensiunea (Voltage):** Valori în timp real primite de la dispozitiv.
- Graficele se actualizează în timp real, iar aplicația reține ultimele 100 de puncte de date.



4. Gestionarea mesajelor de eroare

- Aplicația include o **zonă de erori**, unde sunt afișate notificări pentru:
 - **Conexiune nereușită.**
 - **Erori de citire a datelor.**
 - **Mesaje de succes sau avertizări.**
- Fiecare mesaj este clasificat printr-o culoare specifică:
 - **Succes:** Verde.
 - **Avertisment:** Galben.
 - **Eroare:** Roșu.



5. Funcționalități interactive

- **Deconectarea:** Utilizatorul poate opri conexiunea hardware printr-un buton dedicat. Acesta resetează starea aplicației la modul de așteptare.
- **Scroll pentru consolă:** În secțiunea "Consola Serială", utilizatorul poate vizualiza toate datele brute primite de la dispozitiv, în ordine cronologică.

CAPITOLUL 3. HARDWARE

Placa ESP32-WROOM-32D:

ESP32-WROOM-32D este o placă puternică și versatilă, care integrează funcționalități Wi-Fi și Bluetooth, fiind ideală pentru o varietate de aplicații, de la rețele de senzori cu consum redus de energie până la sarcini complexe.

Specificații principale:

- **Microcontroler:** ESP32-D0WDQ6, un cip dual-core Xtensa® 32-bit LX6, cu frecvență de până la 240 MHz.
- **Memorie:**
 - **RAM:** 520 KB SRAM.
 - **Flash:** 4 MB SPI flash integrat.
- **Conectivitate:**
 - **Wi-Fi:** 802.11 b/g/n.
 - **Bluetooth:** Bluetooth v4.2 BR/EDR și BLE.
- **Tensiune de operare:** 3,3V.
- **Consum de energie:**
 - **Mod activ:** aproximativ 160 mA.
 - **Mod de repaus profund:** aproximativ 10 μ A.

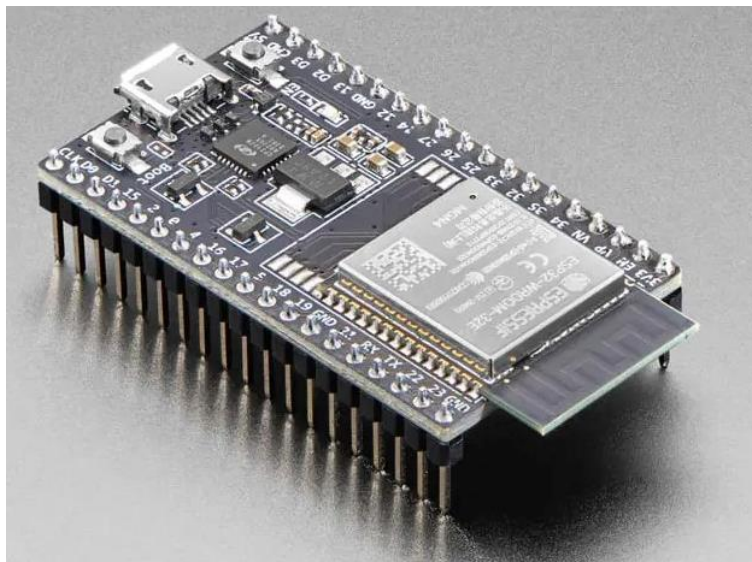


Fig.1. Placa ESP32-WROOM-32D

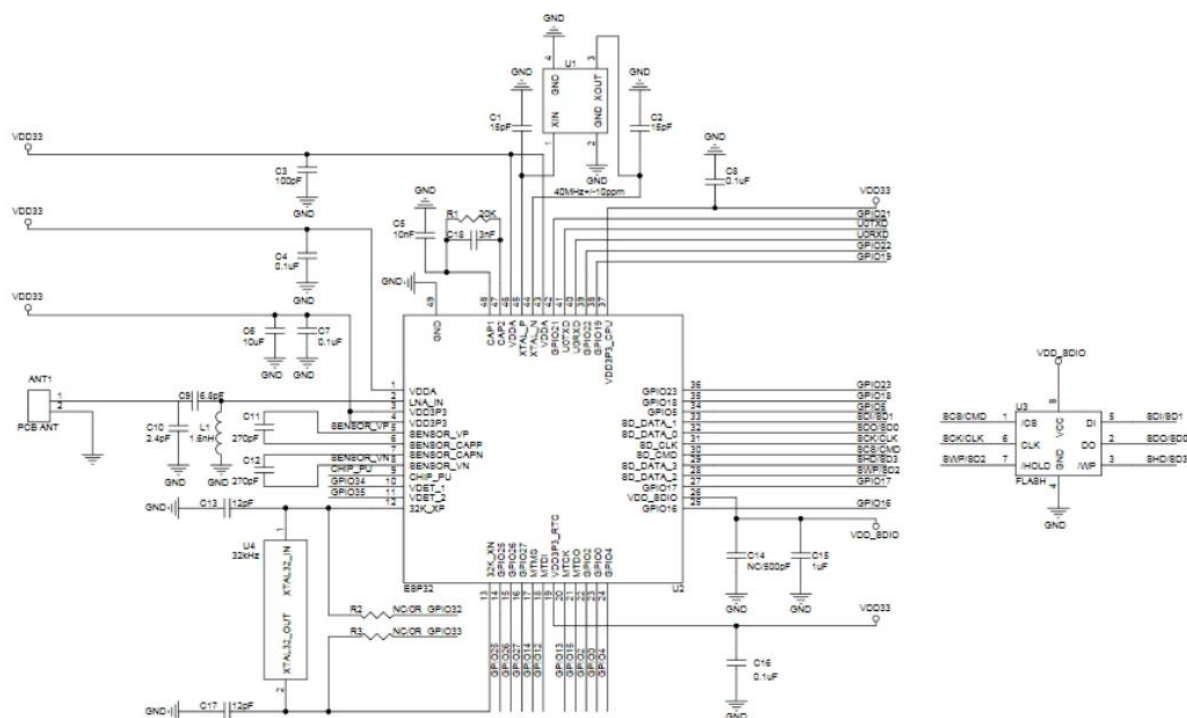


Fig.2. Schema electrică a plăcii ESP32-WROOM-32D

Configurația pinilor:

Placa ESP32-WROOM-32D dispune de 38 de pini, fiecare având funcții multiple așa cum se poate observa din următoarea descriere și din Fig. 6.

Alimentare:

Modulul necesită o tensiune de alimentare de 3,3V.

Este esențial să se asigure o sursă de alimentare stabilă și adecvată pentru a evita deteriorarea modulului. Espressif Systems. „ESP32-WROOM-32D Datasheet”. Espressif Systems, https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32_wroom32u_datasheet_en.pdf, Accesat la 11 ianuarie 20

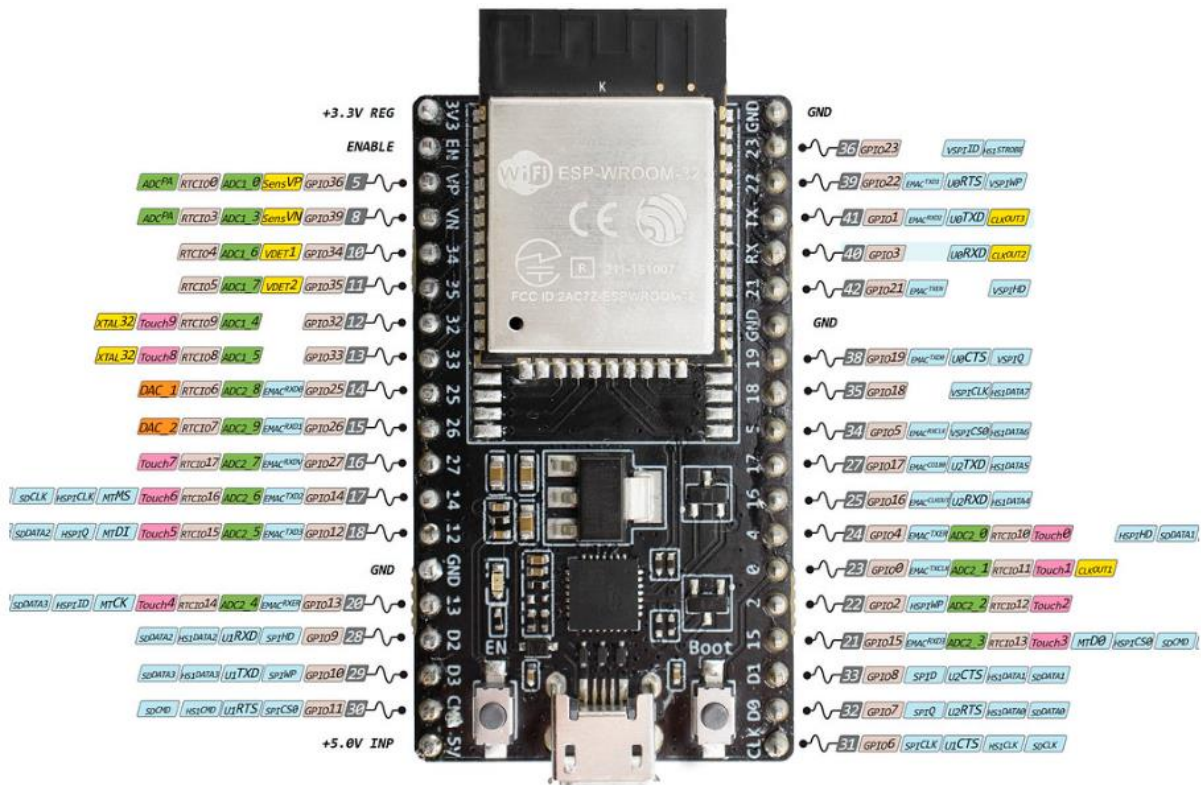


Fig.3. Configurația pinilor ESP32-WROOM-32D

Placa Arduino Uno:

Placa Arduino Uno este o platformă de dezvoltare bazată pe microcontrolerul ATmega328P, concepută pentru a facilita prototipurile electronice și învățarea programării. Aceasta dispune de 14 pini de intrare/ieșire I/O digitali (șase capabili de ieșire PWM), 6 pini I/O analogici și este programabilă folosind mediul de dezvoltare integrat Arduino IDE, printr-un cablu USB de tip B. Tensiunea de operare este de 5V, iar tensiunea de alimentare recomandată variază între 7V și 12V, cu o limită maximă de 20V. Placa include un oscilator ceramic de 16 MHz, un port USB pentru programare și alimentare, un conector de alimentare, un header ICSP și un buton de resetare. Memoria flash disponibilă pentru stocarea codului este de 32 KB, din care 0.5 KB sunt utilizați de bootloader. De asemenea, dispune de 2 KB SRAM și 1 KB EEPROM (ARDUINO DOCUMENTATION, 2024).

ARDUINO DOCUMENTATION, 2024, [Arduino UNO Pinout Complete Guide \[With Diagram\]](#)

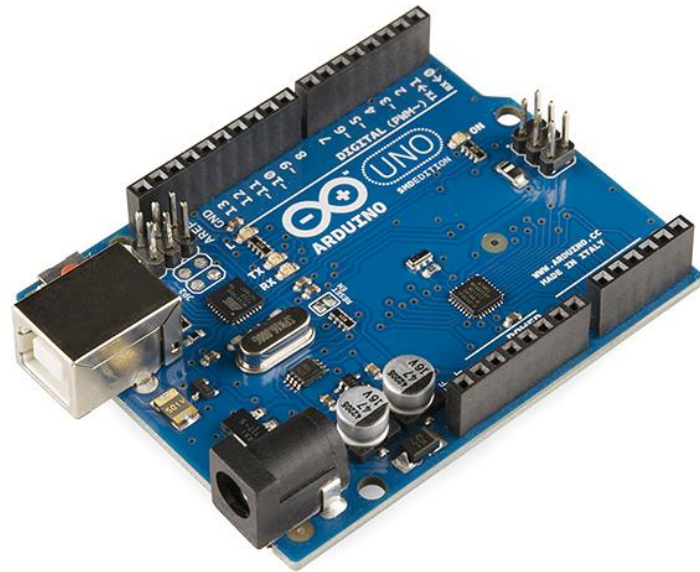


Fig. 4. Placa ARDUINO UNO

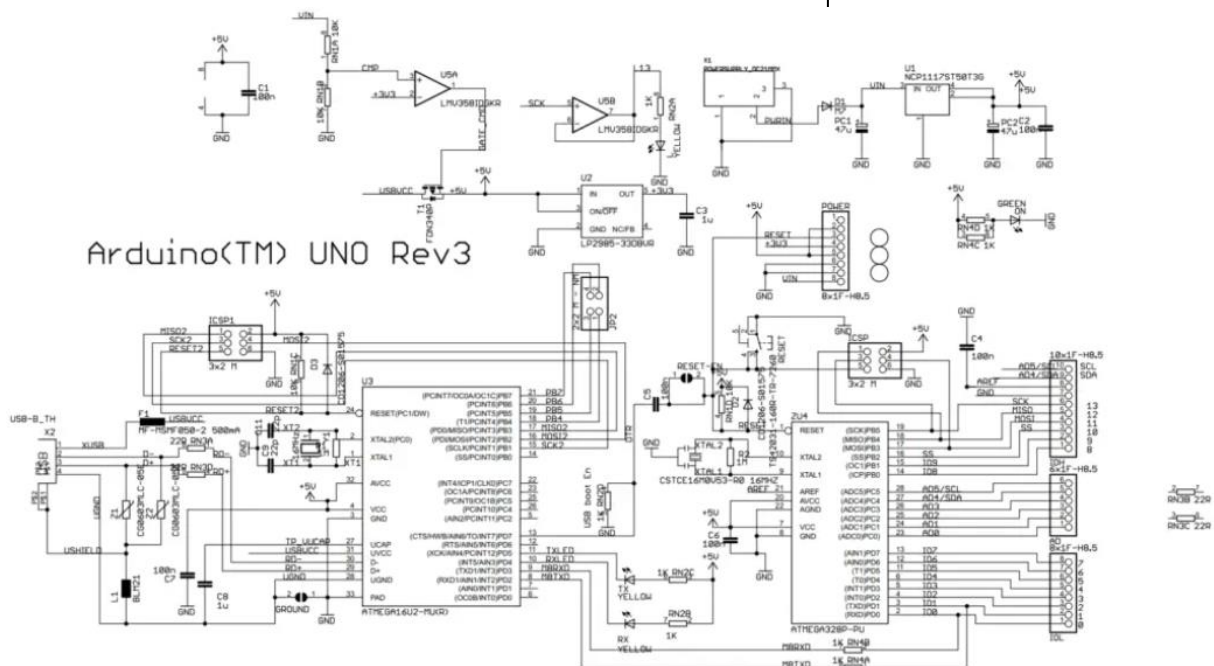


Fig.5. Schema electrică a plăcii ARDUINO UNO

- **Reset:** De obicei folosit pentru a adăuga un buton de reset pe shield-urile care blochează accesul la butonul de reset de pe placă. (ARDUINO DOCUMENTATION, 2024).

Funcții speciale ale pinilor

Fiecare dintre cei 14 pini digitali și 6 pini analogici ai plăcii Arduino Uno poate fi folosiți ca intrare sau ieșire. Acești pini operează la 5 volți. Fiecare pin poate furniza sau primi 20 mA în condiții normale de operare. O valoare maximă de 40 mA nu trebuie depășită pentru niciun pin I/O pentru a evita deteriorarea permanentă a microcontrolerului.

Placa Uno are 6 intrări analogice, etichetate A0 până la A5; fiecare oferă o rezoluție de 10 biți (adică 1024 valori diferite). În mod implicit, acestea măsoară de la masă (GND) până la 5 volți, deși este posibil să modificați limita superioară a intervalului utilizând pinul **AREF** și funcția **analogReference()**.

- **Serial / UART:** Pinii 0 (RX) și 1 (TX). Folosiți pentru a primi (RX) și transmite (TX) date seriale TTL. Acești pini sunt conectați la pinii corespunzători ai cipului ATmega8U2 folosit ca USB-to-TTL serial.
- **Pinii 2 și 3:** Acești pini pot fi configurați pentru a declanșa o întrerupere la o valoare joasă, o tranziție crescătoare sau descrescătoare, sau o schimbare de valoare.
- **PWM (modulație în lățime de impuls):** Pinii 3, 5, 6, 9, 10 și 11. Pot oferi o ieșire PWM pe 8 biți utilizând funcția **analogWrite()**.
- **SPI (Interfață Serială Periferică):** Pinii 10 (SS), 11 (MOSI), 12 (MISO) și 13 (SCK). Acești pini suportă comunicarea SPI utilizând biblioteca **SPI**.
- **I²C (Interfață pe două fire) / I²C:** Pinul **SDA** (A4) și pinul **SCL** (A5). Suportă comunicarea I²C folosind biblioteca **Wire**.
- **AREF (referință analogică):** Tensiunea de referință pentru intrările analogice. *Arduino Uno Pinout, Specifications, Pin Configuration & Programming*. Components101. <https://components101.com/microcontrollers/arduino-uno>

Encoderul HW-040:

Encoderul HW-040 este un encoder rotativ incremental care convertește mișcarea unghiulară a

unui ax într-un semnal electric digital. Acesta este echipat cu patru pini: VCC pentru alimentare, GND pentru masă, DT pentru semnal de date și CLK pentru semnal de ceas. Encoderul HW-040 este utilizat în aplicații care necesită feedback precis al poziției, precum controlul volumului, selecția meniurilor, robotică, control industrial și monitorizarea poziției (CIRKIT DESIGNER, 2024). Un encoder rotativ este un tip de senzor de poziție utilizat pentru determinarea poziției unghiulare a unui arbore rotativ. Acesta generează un semnal electric, analogic sau digital, în funcție de mișcarea de rotație

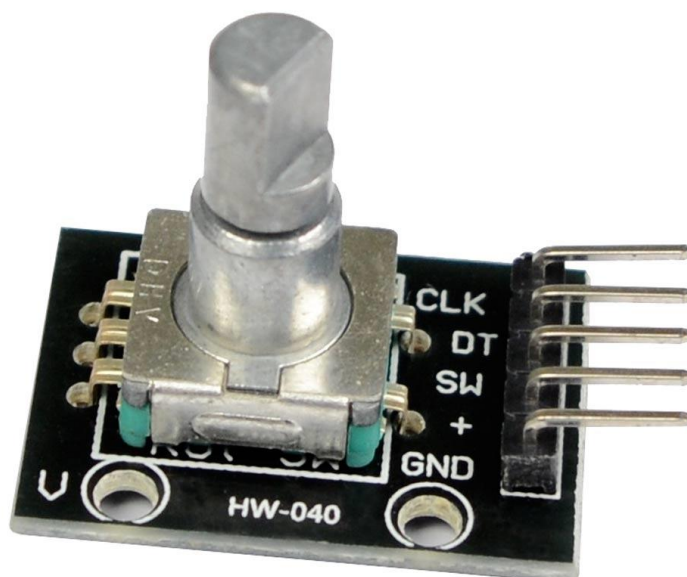


Fig. 7. Encoder HW-040



Fig.8. Prezentarea pinilor Encoderul HW-040

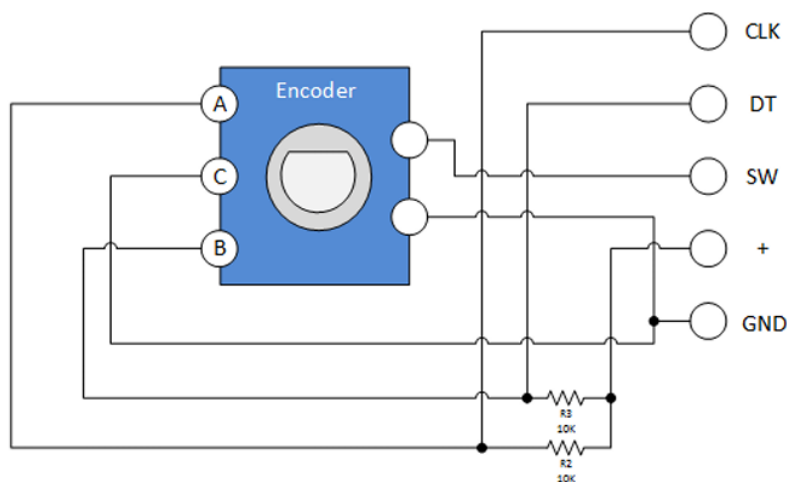


Fig.9. Schema electrică pentru Encoderul HW-040

Conexiunile realizate:

Encoderul monitorizează rotațiile motoarelor și transmite date către placa Arduino. Conexiunile sunt realizate astfel:

- **Pinul DT** al encoderului este conectat la **pinul digital 2** al plăcii Arduino.
- **Pinul CLK** al encoderului este conectat la **pinul digital 3** al plăcii Arduino.
- **Pinul GND** al encoderului este conectat la masa comună.
- **Pinul VCC (+)** al encoderului este conectat la sursa de **5V**

Aceste conexiuni permit Arduino să primească semnale de la encoder, monitorizând astfel rotațiile și direcția de rotație a motoarelor.

Comunicarea între placa Arduino și modulul ESP32-WROOM-32D se realizează prin intermediul unei conexiuni seriale UART (Universal Asynchronous Receiver-Transmitter). Pini 10 și 11 ai plăcii Arduino sunt utilizați pentru această comunicare, unde pinul 10 al Arduino este conectat la pinul 17 al ESP32, iar pinul 11 al Arduino este conectat la pinul 16 al ESP32. Această conexiune permite transferul bidirecțional de date între cele două dispozitive.

În ceea ce privește comunicarea dintre ARDUINO și calculator, placa dispune de interfețe UART hardware care permit stabilirea unei conexiuni seriale directe cu PC-ul. Prin conectarea ARDUINO la calculator prin intermediul unui cablu USB, se poate utiliza monitorul serial sau alte aplicații de terminal serial pentru a transmite și recepționa date.

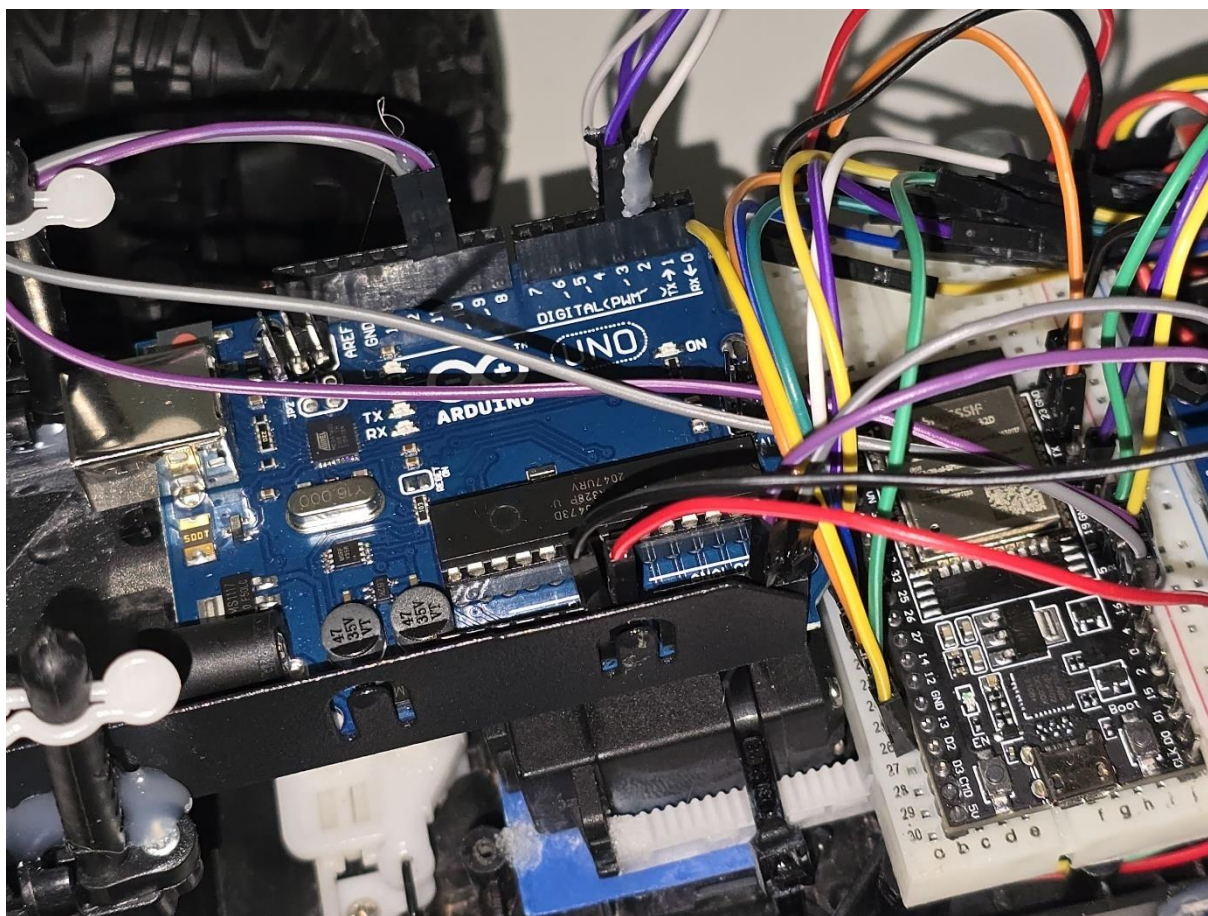


Fig. 10. Conexiunile realizate dintre Arduino Uno și ESP32-WROOM-32D

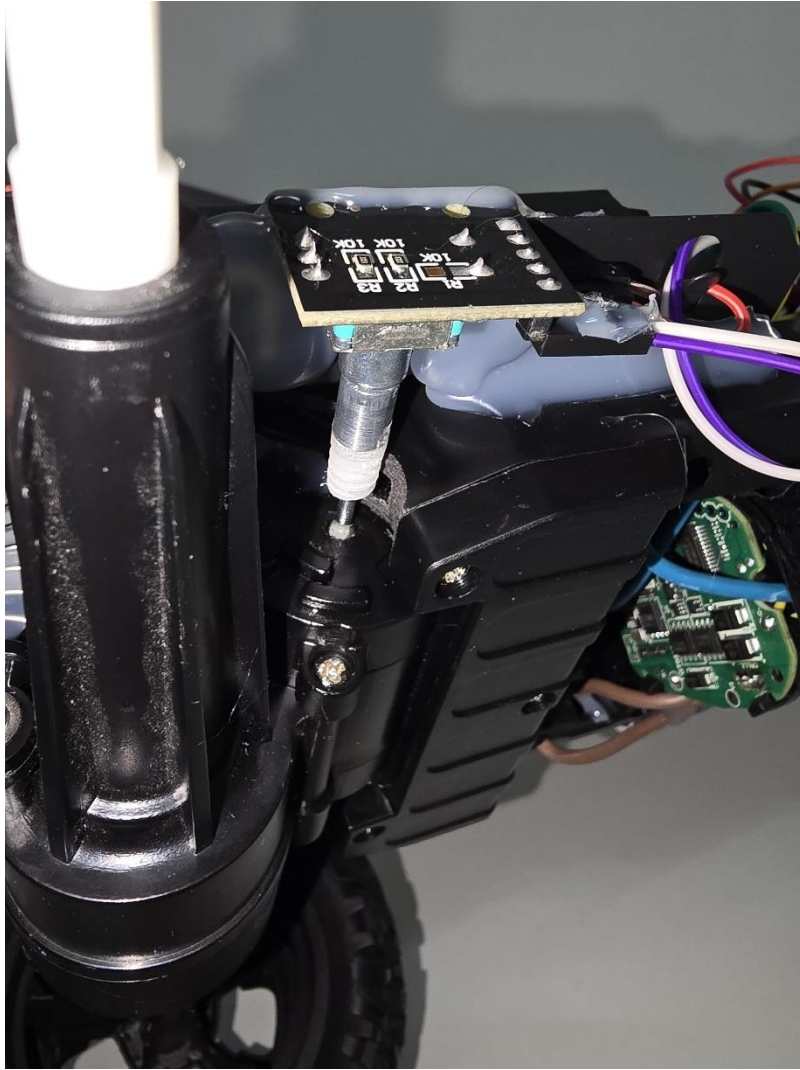
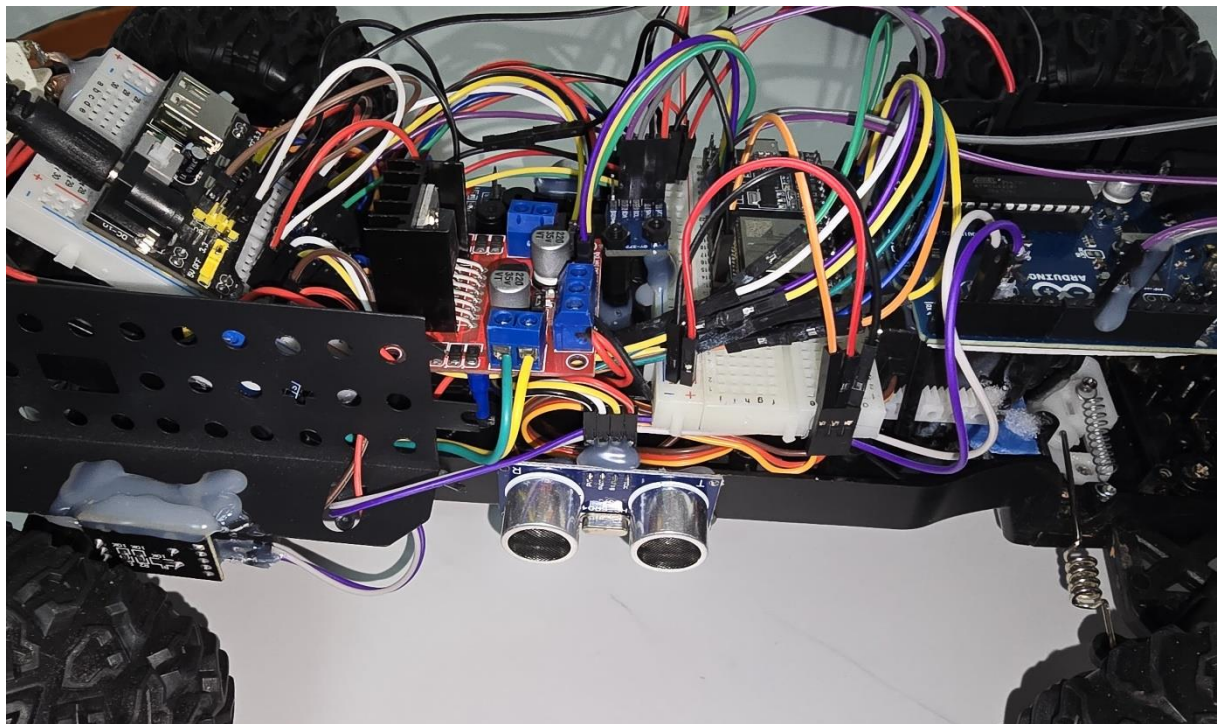


Fig. 11. Conexiunile realizate dintre Arduino Uno și encoder HW-040



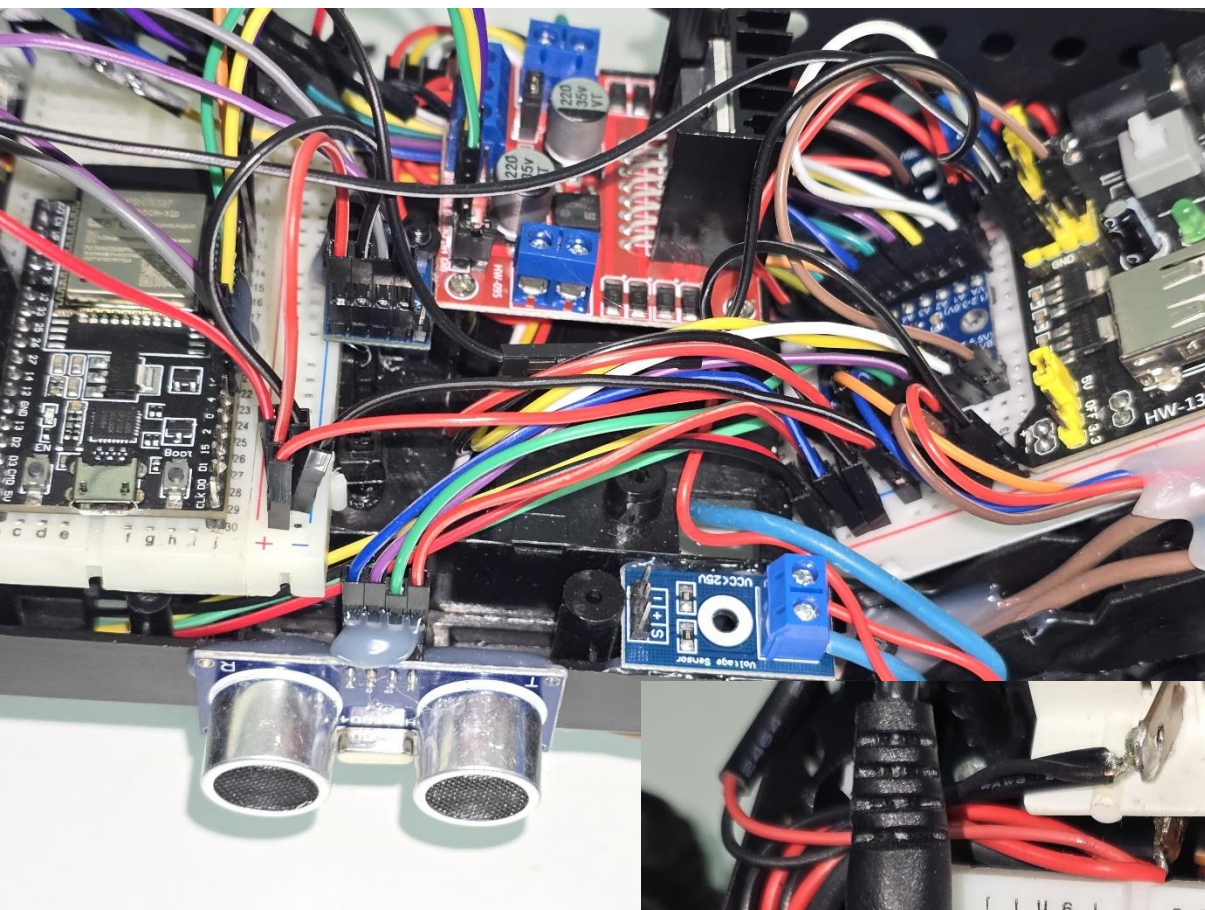
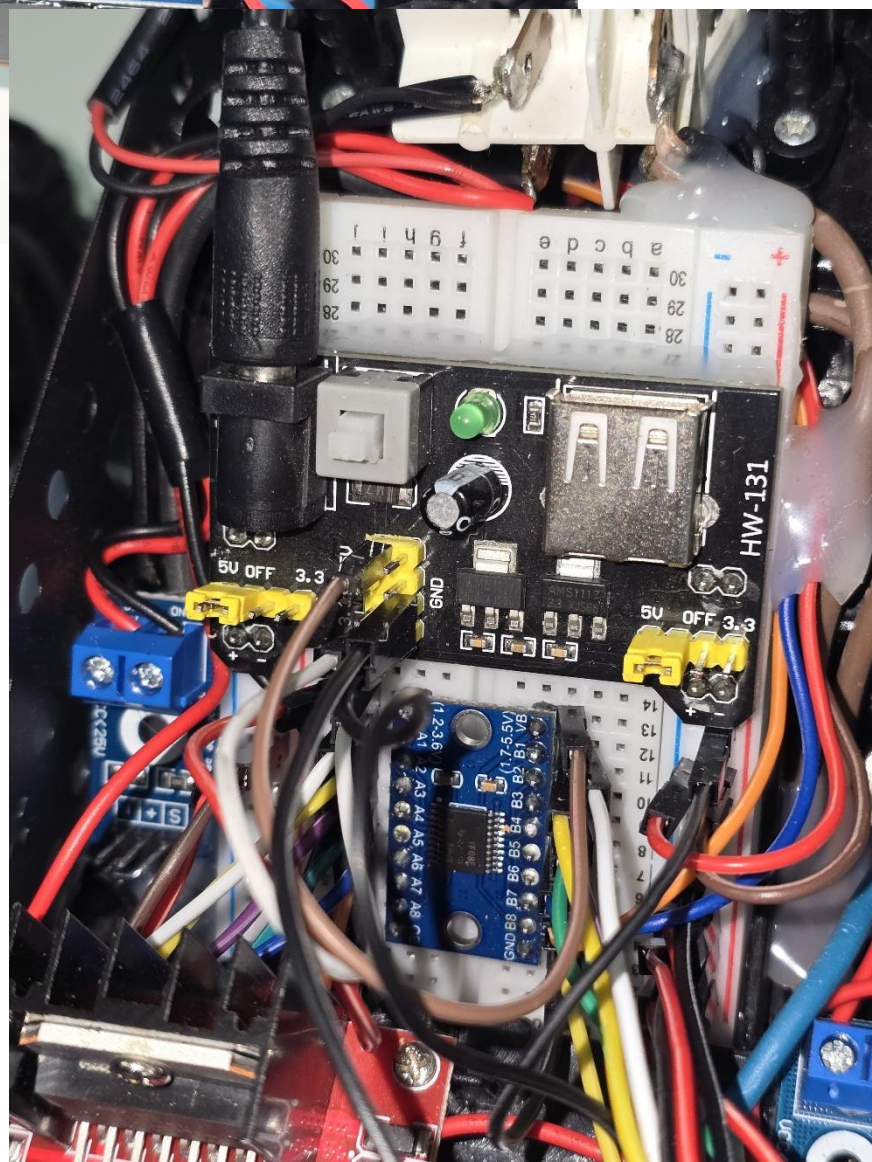


Fig. 12. Conexiunile realizate
dintre senzorul ultrasonic HC-
SR04 și Convertor de nivel logic
TXS0108E



CAPITOLUL 4. CONCLUZII

Proiectul reprezintă o demonstrație practică a integrării dintre hardware și software, utilizând tehnologii moderne pentru monitorizarea și analiza datelor în timp real. Prin intermediul plăcii Arduino și al aplicației web dezvoltate, am reușit să creăm un sistem eficient de urmărire a parametrilor vehiculului, care poate fi utilizat atât în scopuri educative, cât și pentru optimizarea performanțelor vehiculelor reale.

Rezultatele obținute demonstrează următoarele:

1. **Fiabilitate și accesibilitate:**

- Sistemul permite o conectare rapidă și ușoară la placa Arduino pentru citirea datelor în timp real.
- Interfața intuitivă face ca aplicația să fie accesibilă utilizatorilor, indiferent de nivelul de cunoștințe tehnice.

2. **Acuratețea și relevanța datelor:**

- Parametrii esențiali precum RPM, tensiunea bateriei și coordonatele X, Y, Z sunt afișați clar, oferind utilizatorilor informații precise despre starea vehiculului.
- Graficele dinamice îmbunătățesc înțelegerea variațiilor și identificarea eventualelor probleme.

Prin integrarea tehnologiilor IoT, am reușit să dezvoltăm o soluție modernă, care poate avea aplicații variate în educație, cercetare și industrie.

Acest proiect nu doar că oferă un exemplu concret al potențialului tehnologiilor inteligente, dar încurajează și explorarea continuă a soluțiilor inovatoare în domeniul ingineriei și al științei datelor.

ANEXE:

```
ad,
    #23a2f6
);
left: -80px;
top: -80px;
}
.shape:last-child{
    background: linear-gradient(
        to right,
        #ff512f,
        #f09819
    );
    right: -30px;
    bottom: -80px;
}
form{
    height: 520px;
    width: 400px;
    background-color: rgba(255,255,255,0.13);
    position: absolute;
    transform: translate(-50%,-50%);
    top: 50%;
    left: 50%;
    border-radius: 10px;
    backdrop-filter: blur(10px);
    border: 2px solid rgba(255,255,255,0.1);
    box-shadow: 0 0 40px rgba(8,7,16,0.6);
    padding: 50px 35px;
}
```

```
form *{
    font-family: 'Poppins',sans-serif;
    color: #ffffff;
    letter-spacing: 0.5px;
    outline: none;
    border: none;
}
form h3{
    font-size: 32px;
    font-weight: 500;
    line-height: 42px;
    text-align: center;
}

label{
    display: block;
    margin-top: 30px;
    font-size: 16px;
    font-weight: 500;
}
input{
    display: block;
    height: 50px;
    width: 100%;
    background-color: rgba(255,255,255,0.07);
    border-radius: 3px;
    padding: 0 10px;
    margin-top: 8px;
    font-size: 14px;
    font-weight: 300;
}
```

```
::placeholder{
    color: #e5e5e5;
}
button{
    margin-top: 50px;
    width: 100%;
    background-color: #ffffff;
    color: #080710;
    padding: 15px 0;
    font-size: 18px;
    font-weight: 600;
    border-radius: 5px;
    cursor: pointer;
}
.social{
    margin-top: 30px;
    display: flex;
}
.social div{
    background: red;
    width: 150px;
    border-radius: 3px;
    padding: 5px 10px 10px 5px;
    background-color: rgba(255,255,255,0.27);
    color: #eaf0fb;
    text-align: center;
}
.social div:hover{
    background-color: rgba(255,255,255,0.47);
}
.social .fb{
```

```
margin-left: 25px;
}
.social i{
margin-right: 4px;
}
```

JAVA:

```
function validateLogin(event) {
    event.preventDefault(); // Previne trimiterea formularului

    const username = document.getElementById("username").value;
    const password = document.getElementById("password").value;

    if (username === "admin" && password === "admin") {
        // Dacă user și parola sunt corecte, deschide o pagină goală
        window.open('about:blank', '_self');
        window.close();
    } else {
        alert("Username sau parola incorectă. Încearcă din nou.");
    }
}
```