**Smart Parking System**

# **Rolurile fiecărui membru al echipei**

# Deși la începutul proiectului am stabilit o împărțire clară a responsabilităților pentru a eficientiza procesul de lucru, pe parcurs am constatat că implicarea activă și colaborarea strânsă au dus la cele mai bune rezultate. Astfel, fiecare membru al echipei a contribuit în mod semnificativ și echilibrat la toate etapele dezvoltării proiectului – de la scrierea documentației și construcția machetei, până la implementarea și testarea codului de control. Această abordare colaborativă a asigurat o înțelegere comună profundă a funcționării sistemului și o calitate superioară a rezultatului final.

|  |  |  |
| --- | --- | --- |
| **Nume și Prenume** | **Responsabilitate principală** | **Alte contribuții** |
| Dumitrașcu Andrei-Mihnea | Documentație | Participare activă la montaj și testare cod |
| Milu Cătălin-Constantin | Construcție și montaj machetă | Implicare în redactare și validare funcționare |
| Vlăduțu Vlad-Cristian | Cod de control al sistemului | Asistență la montaj și redactare |

1. **Obiectivele proiectului**

Scopul principal al proiectului este dezvoltarea unui sistem inteligent de parcare rotativă, care să eficientizeze utilizarea spațiului urban și să contribuie la reducerea numărului de mașini parcate neregulamentar pe trotuare sau în alte zone neautorizate.

Obiectivele specifice ale proiectului sunt:

* Proiectarea unui sistem de parcare în formă circulară, care să se rotească automat pentru a poziționa un loc liber în dreptul mașinii.
* Implementarea unui sistem de control capabil să detecteze locurile libere și să comande rotația precisă a platformei.
* Integrarea unor funcționalități moderne precum plata automată a parcării.
* Crearea unei machete funcționale pentru demonstrarea conceptului.
* Asigurarea unei soluții scalabile, care poate fi extinsă pe verticală în viitor, pentru a mări capacitatea de parcare fără a ocupa mai mult spațiu la sol.

1. **Descrierea domeniului ales și a soluțiilor similare**

Domeniul abordat este cel al sistemelor inteligente de parcare urbană, o ramură din ce în ce mai importantă în contextul aglomerării orașelor și al creșterii numărului de vehicule. În multe zone urbane, locurile de parcare sunt insuficiente sau greu de accesat, ceea ce duce la parcări ilegale, blocarea trotuarelor și probleme de trafic.

Soluții similare existente:

* **Parcările etajate clasice** – oferă un număr mare de locuri, dar ocupă mult spațiu și implică costuri ridicate de construcție.
* **Sistemele de parcare automatizate verticale** – sunt folosite în anumite orașe mari (ex: Tokyo, New York), dar sunt adesea complexe și greu de întreținut.
* **Parcările inteligente cu senzori** – detectează locurile libere și oferă informații în timp real, dar nu rezolvă problema optimizării spațiului.

Soluția propusă de noi combină avantajele acestor sisteme, oferind o variantă mai compactă și accesibilă: o parcare circulară, rotativă, cu operare simplă, posibilitate de automatizare completă, opțiune de extindere pe verticală.

1. **Componente necesare**

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Componentă** | **Preț** |
| 1. | ATmega328p de pe Placă de Dezvoltare Compatibilă cu Arduino UNO R3 | Componentă proprie |
| 2. | Placă de dezvoltare compatibilă cu Arduino UNO (ATmega328p și CH340) | Componentă proprie |
| 3. | Arduino Uno Rev3 (A000066) | Componentă proprie |
| 4. | Set Motor Pas cu Pas 28BYJ-48 5V și Driver ULN2003 Albastru | 16.97 lei |
| 5. | Hitec HS-7115TH Titanium Thin Wing 7.4V Digital Servo (8mm) | Componentă proprie |
| 6. | Modul RFID RC522 | 9.99 lei |
| 7. | Cartelă RFID 125kHz | 2.99 lei |
| 8. | Convertor de Niveluri Logice Bidirecțional pe 8 Biți TXS0108E | 6.49 lei |
| 9. | Shield cu display TFT cu touchscreen de 2.4’’ | 65.99 lei |
| 10. | Fire Colorate Mamă-Tată (10p) 10 cm | 2.99 lei |
| 11. | Fire Colorate Mamă-Mamă (10p) 20 cm | 2.97 lei |
| 12. | Breadboard HQ (400 Points) | 4.56 lei |
| 13. | Disc imprimat 3D, diametru 6 cm, grosime 0.5 cm | Componentă prorpie |
| 14. | Baterie portabilă 10000 mAh | Componentă proprie |
| 15. | Foamboard alb Airplac, grosime 5 mm, dimensiune 50 x 70 | Componentă proprie |
| **Preț total componente** | | **163.65 lei** |

1. **Descrierea soluției implementate cu prezentarea funcționalităților aferente soluției**

Soluția implementată constă într-un sistem inteligent de parcare rotativă, dezvoltat cu ajutorul platformei Arduino UNO, care gestionează accesul și ieșirea autoturismelor într-o structură circulară. Sistemul folosește un motor pas cu pas pentru a roti platforma de parcare și a poziționa un loc liber în dreptul barierei de acces. Pentru controlul accesului, este utilizat un modul RFID, care identifică utilizatorii autorizați și declanșează deschiderea unei bariere acționate de un servomotor. După ce mașina este parcată sau scoasă, platforma revine automat la poziția inițială sau se rotește pentru a acomoda următoarea solicitare. În momentul ieșirii din sistem, utilizatorul este informat pe ecranul LCD cu privire la durata totală a staționării și costul aferent, calculat automat în funcție de timpul petrecut în parcare.

Proiectul este conceput într-o manieră modulară, astfel încât fiecare componentă majoră este gestionată de un Arduino dedicat. Trei plăci Arduino UNO comunică între ele prin interfață serială, pentru a coordona în mod eficient funcțiile sistemului. Unul dintre Arduino-uri gestionează modulul RFID, ocupându-se de identificarea și validarea accesului. Al doilea Arduino controlează partea mecanică – motorul pas cu pas responsabil de rotația platformei și servomotorul ce acționează bariera. Cel de-al treilea Arduino administrează afișajul LCD, care oferă feedback vizual utilizatorului cu privire la starea sistemului, dar și informațiile privind durata și costul la ieșire.

Această arhitectură modulară asigură nu doar o mai bună organizare a codului și o ușurință în depanare, ci și o scalabilitate ridicată – sistemul poate fi extins în viitor pentru a integra comunicare wireless (Wi-Fi) între module, control de la distanță, monitorizare în timp real sau integrarea într-o aplicație mobilă. Prin această abordare, proiectul propus nu doar rezolvă o problemă practică, ci oferă și o bază solidă pentru dezvoltări ulterioare inteligente și eficiente.

Funcționalitățile principale sunt:

1. **Identificarea utilizatorilor cu ajutorul cartelelor RFID**

Pentru a intra sau ieși din parcare, utilizatorul apropie cartela RFID de cititorul MFRC522. Sistemul identifică UID-ul cartelei și decide dacă utilizatorul este:

* **nou (intrare)**: se caută un loc liber și se înregistrează ora de intrare;
* **existent (ieșire)**: se calculează durata staționării și costul.

**Fragment de cod – citire UID RFID:**

if (!mfrc522.PICC\_IsNewCardPresent()) return;

if (!mfrc522.PICC\_ReadCardSerial()) return;

String uid = "";

for (byte i = 0; i < mfrc522.uid.size; i++) {

if (mfrc522.uid.uidByte[i] < 0x10) uid += "0";

uid += String(mfrc522.uid.uidByte[i], HEX);

}

lot.processCard(uid);

1. **Rotația platformei circulare către locul disponibil sau ocupat**

Sistemul utilizează un motor pas cu pas 28BYJ-48 pentru a roti platforma în funcție de locul țintă. Fiecare spațiu este indexat, iar rotația se face în sensul cel mai scurt.

**Fragment de cod – rotație automată:**

int diff = (targetIndex - currentIndex + NO\_SPACES) % NO\_SPACES;

if (diff > NO\_SPACES / 2) {

stepper.step(-stepsPerSpace \* (NO\_SPACES - diff));

} else {

stepper.step(stepsPerSpace \* diff);

}

1. **Deschiderea și închiderea automată a barierei**

Barierea acționată de un servo-motor se ridică automat atunci când o mașină trebuie să intre sau să iasă din sistem. Mișcarea este controlată gradual pentru un efect realist.

**Fragment de cod – deschiderea barierei:**

for (int angle = closedAngle; angle <= openAngle; angle++) {

servo.write(angle);

delay(15);

}

delay(5000); // așteaptă ca mașina să treacă

for (int angle = openAngle; angle >= closedAngle; angle--) {

servo.write(angle);

delay(15);

}

1. **Calcul automat al timpului de staționare și al costului**

La ieșirea din parcare, sistemul calculează timpul petrecut de mașină pe baza diferenței între millis() actual și timpul de intrare, aplicând un tarif per minut.

**Fragment de cod – calcul durată și cost:**

unsigned long durationMillis = millis() - entryTime;

float minutes = durationMillis / 60000.0;

float price = minutes \* pricePerMinute;

Serial.print("Time: "); Serial.print(minutes, 2);

Serial.print(" min. Price: $"); Serial.println(price, 2);

1. **Testarea soluției**

Testarea soluției s-a realizat pe parcursul dezvoltării, folosind mai multe cartele RFID și scenarii posibile:

|  |  |  |
| --- | --- | --- |
| **Scenariu testat** | **Rezultat așteptat** | **Rezultat obținut** |
| Cartea RFID este nouă | Bariera se deschide, platforma rotește la loc liber | Confirmat |
| Cartea RFID este recunoscută | Platforma rotește, bariera se ridică, se calculează costul | Confirmat |
| Nu mai sunt locuri libere | Se afișează mesajul „No empty parking spaces” | Confirmat |
| Platforma este deja la poziția dorită | Nu se face rotație inutilă | Confirmat |

Pentru simulare, a fost utilizată consola serială, cu UID-uri introduse manual sau prin cartela fizică. Funcționarea mecanică a platformei a fost verificată pe macheta construită, iar mișcarea barierei a fost testată în buclă.

1. **Referințe**

* <https://www.optimusdigital.ro/ro/>
* <https://www.arduino.cc/>