

PARCARE INTELIGENTĂ

PROIECT REALIZAT DE:

GIUROIU DARIUS-LUCIAN

CURCĂ ȘTEFAN

IONIȚĂ THEODOR

CUPRINS

- Contribuțiile fiecărei persoane din proiect
- Obiectivele proiectului propus
- Descrierea domeniului ales și a soluțiilor similare
- Descrierea soluției propuse
- Descrierea soluției implementate cu prezentarea funcționalităților aferente soluției
- Testarea soluției

a) Contribuțiile fiecărei persoane din proiect

Curcă Ștefan : Responsabil de cercetare, implementarea hardware și software a proiectului

Giuroiu Darius-Lucian: Responsabil de cercetare, redactarea documentației și implementare software

Ioniță Theodor: Responsabil de redactarea documentației și implementarea hardware

b) Obiectivele proiectului propus

Prin acest proiect ne propunem să îmbunătățim accesul în parcarile instituțiilor, hotelurilor, mall-urilor, cartierelor rezidențiale și a tuturor parcărilor private prin implementarea unui model sigur și eficient de barieră automată ce detectează autovehiculele care intră, respective care ies din parcare.

Un obiectiv al proiectului ar fi controlul fluxului de vehicule. Prin intermediul barierei automate, se poate controla fluxul de vehicule în și din parcare respectivă. Aceasta poate permite o administrare mai eficientă a spațiului disponibil, asigurând că numărul de vehicule în parcare respectivă nu depășește capacitatea acestuia.

Un alt obiectiv ar fi eficientizarea din punct de vedere al timpului. Vom elimina necesitatea angajării unui gardian care va deschide bariera, aceasta putând funcționa neîntrerupt pe toată durata zilei.

Ne dorim ca soluția noastră să fie una simplă, rapidă, eficientă, fezabilă și orientată către nevoile conducătorilor auto.

c) Descrierea domeniului ales și a soluțiilor similare

Domeniul ales de către noi este cel al parcărilor “automate”. Ideea propusă de noi nu este chiar o parcare automată dar face parte din sistemul uneia, așadar noi ne-am ocupat doar de accesul în parcare, de aici mai departe domeniul se extinde semnificativ în zona de optimizare a spațiului și cea de securitate.

Una din problemele marilor orașe este aceea a parcărilor pentru vehicule. De-a lungul timpului s-au încercat diferite metode pentru amenajarea acestor parcuri. De la parcurile obișnuite, la cele subterane, la cele supraetajate până la parcurile “automate”. Toate aceste tipuri de parcuri au nevoie de o formă controlată de acces și de ieșire așa că s-au implementat mai multe metode prezentate mai jos.

Soluții similare:

- Accesul în parcare pe baza unei cartele: Pentru îmbunătățirea securității o soluție ar putea fi accesul în parcare prin intermediul unei cartele de acces. Aceasta poate fi o cartela magnetică, o cartelă RFID (Radio Frequency Identification) sau o cartelă cu cip. Sistemul citește informațiile de pe cartelă și, în funcție de valabilitatea și permisiunile acesteia, ridică bariera pentru a permite intrarea.
- Sistem de recunoaștere a plăcilor de înmatriculare: Aceste sisteme utilizează camere speciale pentru a citi și recunoaște automat numerele de înmatriculare ale vehiculelor care intră și ies din parcare. Aceasta facilitează monitorizarea utilizării parcarilor și identificarea vehiculelor neautorizate sau a celor care încalcă regulile parcarilor. Acest tip de parcare inteligentă oferă adesea opțiuni de plată și rezervare online, eliminând necesitatea manipulării fizice a biletelor de parcare sau a monedelor. Utilizatorii pot rezerva și plăti pentru locurile de parcare în avans, utilizând aplicații mobile sau alte platforme online.
- Smart Parking este un sistem mecanizat eficient din punct de vedere al spațiului și al costurilor. Acest sistem poate mări capacitatea parcarii de până la 8 ori mai mult. Sistemul este bazat pe o tehnologie care extinde capacitatea parcarii pe verticală pentru eliberarea terenului scump și insuficient. Ocupând doar spațiul pentru 2 parări de mașini (aproximativ 30 mp), unde se pot parca până la 16 mașini.
- Integrare cu alte servicii și infrastructură: Parcarile inteligente pot fi integrate cu alte servicii și infrastructură, cum ar fi aplicații mobile pentru navigație, sisteme de transport public sau stații de încărcare pentru vehiculele electrice.

d) Descrierea soluției propuse:

O parcare inteligentă este un concept care implică utilizarea tehnologiilor și a sistemelor avansate pentru a îmbunătăți gestionarea și funcționarea parcarilor. Scopul unei parări inteligente este de a oferi o experiență mai eficientă și convenabilă pentru utilizatori, de a optimiza utilizarea spațiului de parcare și de a spori siguranța și securitatea în parcare.

Parcarile inteligente utilizează tehnologii precum senzori pentru a detecta și monitoriza locurile de parcare disponibile. Aceste informații sunt apoi afișate în timp real pentru șoferi, permițându-le să găsească rapid un loc de parcare liber.

Soluția noastră propune utilizarea unor senzori IR amplasați la intrarea, respectiv la ieșirea din parcare, pentru identificarea autovehiculelor și doi senzori de proximitate pentru a împiedica bariera să coboare în cazul în care mașina nu a reușit să intre sau să iasă cu succes.

Vom avea amplasat la intrarea în parcare un ecran care va informa șoferii cu privire la numărul locurilor libere. În cazul în care parcare este plină șoferii vor fi notificați și nu li se va mai permite accesul.

Domenii pentru implementarea soluției:

Putem lua ca exemplu o parcare privată ce aparține unei instituții (hotel, mall, sală de fitness, etc.) care utilizează o barieră clasică gestionată manual de un portar.

Această soluție nu este eficientă în primul rând din punct de vedere al timpului. Sunt mulți clienți care vor ieși și intra din parcare într-un timp foarte scurt ceea ce va duce la o supra-aglomerare la intrare, respectiv la ieșire. Acest lucru este amplificat și de faptul că un portar nu poate fi prezent non-stop. Această problemă poate fi rezolvată prin implementarea unei soluții de barieră cu deschidere automată care nu necesită prezența unui portar și care poate funcționa neîntrerupt pe o perioadă mai lungă de timp fără necesitatea unei revizii tehnice.

Din punct de vedere al securității, pentru a nu permite intrarea în parcare a oricui, instituția poate veni cu o soluție adițională cum ar fi implementarea unui cititor de cartele pentru a verifica identitatea clientului sau a unei sonerii care va notifica recepția.

Această soluție combinată cu a noastră poate menține evidența numărului de clienți într-o anumită perioadă de timp pentru a face un studiu cu privire la profitabilitatea afacerii.

În cazul în care parcare este plină, pe ecranul de la intrare poate apărea un mesaj care va redirecționa clienții către o altă zonă special amenajată unde își pot lăsa mașina, astfel îmbunătățim traficul în zona respectivă.

Implementare unei parcări automate prevede un avantaj considerabil din punct de vedere economic. Achiziționarea unor senzori infraroșu cat și a unor senzori de proximitate este o soluție eficientă și ieftină.

Acești senzori prezintă un consum redus de energie electrică și sunt rezistenți în timp, aceștia având nevoie doar de o revizie tehnică anuală pentru a preveni posibilitatea apariției unei erori tehnice, astfel menținând fiabilitatea sistemului.

e) Descrierea soluției implementate cu prezentarea funcționalităților aferente soluției

Pentru o analiză cât mai exactă a modelului propus am realizat o machetă ce reprezintă o parcare unde urmează să montăm sistemul de bariere implementat în soluția propusă. Sistemul este format din două bariere, una dintre acestea este se află la intrarea în parcare iar cealaltă barieră se află la ieșirea din parcare.

Fiecare barieră dispune de doi senzori pentru detectarea obiectelor (autovehiculelor), în cazul nostru am ales să folosim un modul senzor infraroșu pentru evitarea obstacolelor și un senzor ultrasonic de distanță.

Senzorul de obstacole se bazează pe reflexia radiației infraroșu de către obstacol. Radiația infraroșie este emisă de către un LED și este recepționată de către o fotodiodă. Output-ul acestui senzor este digital iar distanța de detecție poate fi reglată dintr-un potențiometru.

Senzorul ultrasonic emite unde la 40 KHz și măsoară timpul de întoarcere, la ieșire emite un semnal cu amplitudinea semnalului la care este alimentat. Măsurând timpul emiterii semnalului high, se determină distanța, mai exact se folosește formula de calcul: Distanța = (timpul de nivel high * viteza sunetului) / 2.

Pentru acționarea barierelor am folosit două servomotoare de tip FS5103B cu un unghi de rotație de 180° și cu o viteză de 0.17 secunde/60°. Pentru afișarea numărului de locuri rămase disponibile am folosit un LCD ce poate afișa caractere pe două linii și șaisprezece coloane, acest LCD comunică cu placa de dezvoltare prin intermediul unei interfețe I2C.

Pentru colectarea datelor de la senzori, controlul motoarelor și afișarea datelor am folosit o placă de dezvoltare Arduino UNO. Aceasta este o platformă hardware open-source ce are un microcontroler Atmega328P la bază și este echipat cu mai multe porturi digitale și analogice care permit conectarea și controlul diferitelor componente electronice.

Placa vine cu un bootloader preîncărcat, ceea ce înseamnă că poate fi programată fără a avea nevoie de un programator extern. Cu ajutorul software-ului Arduino IDE (Integrated Development Environment), putem scrie și încărca programe pe placa Arduino UNO. Limbajul de programare folosit este un subset al limbajului C/C++, iar IDE-ul oferă o gamă largă de funcții și biblioteci ușor de utilizat, care facilitează dezvoltarea de proiecte interactive și controlul componentelor electronice.

Deoarece placa de dezvoltare are doar doi pini de alimentare și trei pentru ground am folosit un breadboard pentru alimentarea senzorilor și a LCD-ului. Tensiunea de lucru a plăcii este de 5 V așa că se conectează la portul USB al calculatorului/laptopului folosind un cablu de tip USB A-B, poate fi alimentată și extern folosind un alimentator extern. Alimentarea externă este necesară în situația în care consumatorii conectați la placă necesită un curent mai mare de câteva sute de miliamperi, în cazul nostru curentul de ieșire de aproximativ 40 mA este suficient.

Servomotoarele folosite au un cuplu maxim de 3,8 kg · cm și acceptă tensiuni cuprinse între 4,8 și 6 V, pentru prevenirea folosirii de către cele două motoare a unui curent mai mare

decât cel de ieșire al plăcii, fapt ce ar conduce către afectarea măsurătorilor realizate de către senzori, motoarele vor fi alimentate folosind un alimentator extern conectat la rețeaua de 230 V.

Pinul de ground a celor două motoare va fi conectat la pinul de ground comun pentru tot sistemul. Pinii de comandă a celor două motoare vor fi conectate la pinii digitali șase și respectiv șapte ale plăcii de dezvoltare. Comanda inițială pentru cele două servomotoare va fi de 90° pentru setarea poziției de închis a barierei. Pentru ridicarea barierei un servomotor va primi comanda de 10° iar celălalt motor va primi comanda de 180°. Aceste comenzi vor fi date doar dacă senzorii detectează un obstacol iar în cazul motorului amplasat la bariera de la intrarea în parcare se mai adaugă condiția ca numărul de locuri disponibile din parcare să fie mai mare decât zero.

Modulul senzorului infraroșu se alimentează la o tensiune cuprinsă între 3 și 5 V și are un unghi de observare a obstacolelor de 35°. Cu ajutorul potențiometrului montat pe modul putem modifica nivelul de referință, astfel încât să ajustăm sensibilitatea, deci, distanța la care modulul detectează obstacole. Comparatorul furnizează la ieșire 1 logic atunci când nu detectează obstacole și 0 logic atunci când întâlnește un obstacol. Modulul conține și două LED-uri indicatoare pentru alimentare și pentru semnalizarea obstacolelor. Pinii VCC ai modulelor se vor conecta în porturile aflate pe breadboard conectate la rândul lor la pinul de 5V al plăcii de dezvoltare. Pinii de ground ale senzorilor vor fi conectate la porturile breadboard-ului rezervate pentru ground. Pinii de ieșire vor fi conectați la pinii digitali de pe Arduino și anume pinul 2 respectiv pinul 4.

Chiar dacă ridicarea barierelor este determinată inițial de senzorii infraroșii amplasați înaintea barierelor, poate exista cazul în care autovehiculul din diferite cauze să rămână sub barieră și să nu părăsească zona respectivă până la coborârea acesteia, așa că sistemul nostru încorporează doi senzori ultrasonici amplasați pe direcția barierei și care au rolul de a menține bariera ridicată cât timp sub barieră se află un obstacol.

Tensiunea de alimentare a senzorilor are valoarea de 5 V, un curent consumat de 15 mA, un unghi de măsurare de 15° și o eroare de doar 3 mm. Pinii VCC ai senzorilor vor fi conectați pe breadboard în porturile destinate tensiunii de alimentare de 5 V iar pinii de ground vor fi conectați pe breadboard în porturile destinate ground-ului. Pinii Echo și Trig sunt utilizați pentru comunicarea cu senzorul și măsurarea distanței în baza principiului ecoului. Pinul Trig (Trigger) este utilizat pentru a declanșa emisia ultrasunetelor. Atunci când acesta primește un semnal de nivel high (5 V), senzorul emite un impuls ultrasonic. Pinul Echo este utilizat pentru a detecta și măsura timpul care a trecut între emiterea impulsului și primirea ecoului de la obiectul reflector. Pinul Echo generează un semnal de nivel high (5 V) pentru o durată proporțională cu timpul scurs între emiterea și recepționarea ecoului. Pentru primul senzor pinul Trig va fi conectat la pinul digital 11 al plăcii de dezvoltare și pinul Echo va fi conectat la pinul digital 12 iar pentru al doilea senzor pinul Trig va fi conectat la pinul digital 10 și pinul Echo va fi conectat la pinul digital 13.

Procedura de măsurare a distanței decurge astfel: pinul Trig este inițializat la nivelul high (5 V) pentru o perioadă scurtă de timp (de aproximativ 10 μ s), această perioadă declanșează senzorul pentru a emite un impuls ultrasonic. Senzorul emite un impuls ultrasonic și începe să măsoare timpul, dacă există un obiect în fața senzorului, undele ultrasonice vor fi reflectate înapoi către senzor. Senzorul detectează ecoul și pinul Echo trece la nivelul high (5 V), durata în care pinul Echo este la nivelul high este măsurată cu ajutorul unui timer în microsecunde. Pe baza timpului măsurat, distanța este calculată folosind formula prezentată anterior.

Ultima componentă a sistemului este ecranul LCD care este montat la intrarea în parcare și care afișează numărul disponibil de locuri. Pinul VCC al LCD-ului va fi conecta la breadboard în portul destinat tensiunii de alimentare de 5 V iar pinul de ground va fi conectat la breadboard în portul destinat groundului. Pinul SDA al LCD-ului este pinul de date serial și este utilizat pentru transmiterea și recepționarea datelor. Informațiile sunt transferate într-un mod sincronizat, bit cu bit. Pinul SDA va fi conectat la pinul analogic 4 al plăcii de dezvoltare. Pinul SCL este pinul de ceas serial și este utilizat pentru a sincroniza transmiterea și recepționarea datelor, acest semnal de ceas este generat de un dispozitiv master I2C. Pinul SCL va fi conectat la pinul analogic 5 al plăcii de dezvoltare.

Descrierea codului implementat

În prima parte am inclus bibliotecile necesare pentru ecranul LCD “[LiquidCrystal_I2C.h](#)” și pentru cele două servomotoare “[Servo.h](#)”. Am definit pinii pentru senzorii de proximitate și pentru cei cu infraroșu și variabilele unde i-am stocat. Am creat variabilele care vor memora valorile colectate de la cele două tipuri de senzori.

În funcția “void setup()” am inițializat și pregătit fiecare componentă pentru a putea fi utilizată:

- am setat pinii senzorilor;
- am aprins ecranul LCD și am afișat mesajul inițial;
- am setat servomotoarele în poziția de 90° (barieră închisă);

În funcția “void loop()” am pregătit senzorii de proximitate pentru a lua măsurători în timp constant pe toată durata funcționării programului, măsurători pe care le-am afișat în consolă pentru a putea monitoriza funcționarea optimă a senzorilor, și poziționarea ideală a acestora pe planșă pentru a detecta autovehiculele:

```
Serial.print("Distance = ");  
Serial.print(distance);  
Serial.println(" cm");
```


Pasul următor al programului a constituit conceperea unor comenzi logice pentru deschiderea și închiderea barierei. Astfel, când senzorul IR detectează mișcare ”digitalRead(IR1)==LOW” vom apela funcția “`myservo.write(10)`” care va ridica brațul motorului (barieră deschisă) pentru a permite accesul mașinilor în parcare.

De asemenea numărul locurilor libere din parcare va scădea cu o valoare “Slot = Slot – 1”, lucru care va fi afișat și pe ecranul LCD.

```
lcd.setCursor(0,0);  
lcd.print("  WELCOME!  ");  
lcd.setCursor(0,1);  
lcd.print("Slot Left: ");  
lcd.print(Slot);
```

În cazul în care parcare este plină bariera nu se va deschide iar pe ecran va apărea un mesaj care va informa oamenii de acest lucru:

```
lcd.setCursor(0,0);  
lcd.print("  Scuze  ");  
lcd.setCursor(0,1);  
lcd.print("  Parcare plina  ");  
delay(3000);  
lcd.clear();
```

Gândindu-ne la alte soluții similare de parcare automată am observat că majoritatea nu au inclus o soluție pentru cazul în care mașina a rămas blocată sub barieră, lucru care poate cauza atât defectarea barierei cât și o ușoară avariere a mașinii.

Astfel, pe lângă acest senzor IR am optat și pentru includerea unui senzor de proximitate amplasat sub barieră care în momentul când va detecta o mașină, iar numărul locurilor de parcare nu a ajuns la valoarea 0, va menține bariera deschisă pentru a nu coborî peste aceasta. Când senzorul va indica valori mai mari de 15cm se înțelege că mașina a intrat cu succes în parcare și bariera va coborî.

Acest procedeu prezentat mai sus a fost repetat și pentru ieșirea din parcare, cu diferența că de această dată numărul locurilor libere va crește “Slot = Slot + 1”.

În timpul rulării programului am observat că bariera încearcă să se ridice încontinuu când o mașină ajunge în dreptul senzorului IR chiar dacă aceasta este în poziție verticală. Acest lucru se datorează faptului că în codul scris de noi de fiecare dată când senzorul detectează mișcare se

apelează funcția “`myservo.write(10)`”, iar numărul de locuri libere din parcare scade foarte rapid pana ajunge la 0.

Pentru a soluționa această problemă am adăugat o variabilă “`flag1 = 0`” pentru senzorul IR de la intrarea în parcare și variabila “`flag2 = 0`” pentru senzorul de la ieșirea din parcare. Am adăugat această variabilă în condiția pentru ridicarea barierei “`if(digitalRead(IR1)==LOW && flag1==0)`”. Odată ce această condiție a fost îndeplinită variabila “`flag1`” va lua valoarea 1, urmând să fie din nou inițializată cu 0 în momentul în care mașina se va afla în dreptul senzorului de proximitate.

```
if(distance < 15 && Slot > 0)
{
    currentCommand = 10;
    flag1=0;
}
```

Prin aceste condiții problema a fost rezolvată deoarece funcția de ridicare a barierei va fi apelată o singură dată pentru prima citire făcută de senzor.

f) Testarea soluției

Înainte de implementarea fizică a soluției propuse am simulat funcționarea acesteia (imaginea 1) cu ajutorul platformei online Tinkercad. Acesta este un instrument de modelare 3D și CAD (Computer-Aided Design) accesibil și ușor de utilizat. Tinkercad oferă o varietate de instrumente și funcții pentru asamblarea componentelor, manipularea obiectelor și crearea de modele 3D. În urma simulării în Tinkercad a urmat implementarea fizică a soluției, iar după finalizarea acesteia a urmat testarea soluției conform următoarelor situații :

Toate locurile (două în cazul machetei) din parcare sunt libere, ambele bariere sunt coborâte iar ecranul afișează numărul de locuri disponibile. Primul autovehicul se apropie de intrarea în parcare și este detectat de senzorul infraroșu care transmite datele către microcontroler, iar acesta la rândul său transmite comanda de ridicare către barieră și decrementează numărul de locuri disponibile ulterior afișând numărul actualizat de locuri disponibile. Barierea nu va primi comanda de coborâre până când mașina nu va trece de zona barierei. Ulterior barierea va fi coborâtă iar autovehiculul va fi parcat. În cazul în care o a doua mașină se va apropia de intrarea parcării procesul va fi identic.

În cazul în care o altă mașină se apropie de intrarea parcării senzorul infraroșu o detectează și transmite datele către microcontroler, acesta verifică dacă numărul de locuri disponibile este mai mare decât zero, rezultatul fiind unul negativ motorul nu transmite nici o

comandă către motorul barierei, în schimb va transmite o comandă către display pentru afișare mesajului: “Scuze, parcare este ocupată”.

În situația în care un autovehicul dorește să părăsească parcare acesta se va îndrepta în direcția barierei destinate ieșirii din parcare. În apropierea acesteia senzorul infraroșu va detecta mașina și va transmite datele către microcontroler care va transmite comanda de ridicare a barierei motorului și va incrementa numărul de locuri disponibile din parcare. Bariera nu va fi coborâtă cât timp senzorul de proximitate aflat pe direcția acesteia va transmite date microcontrolerului precum că mașina se află în raza de acțiune a barierei. În momentul în care mașina a trecut de barieră, motorul acesteia primește comanda de coborâre și o altă mașină poate primi accesul în parcare.

