Parcare supraetajată cu barieră și lift

Șerban Dragoș-Andrei, 332CB

Introducere

- Proiectul meu constă într-o parcare supraetajată ce utilizează o barieră controlată prin senzori de obstacole și un lift controlat prin butoane. Senzorii de obstacole sunt folosiți pentru a detecta mașinile atunci când vor să intre sau să iasă din parcare, iar butoanele controlează mișcarea liftului și nivelul la care urcă sau coboară.
- Scopul proiectului este de a optimiza utilizarea spațiului de parcare. Prin intermediul sistemului de tip elevator, este optimizat spațiul folosit, iar bariera ce se ridică automat ajută la reducerea timpului pentru utilizatorii parcării și contorizează câte mașini sunt prezente în parcare în fiecare moment.
- Am pornit de la ideea că oamenii folosesc mașina personală din ce în ce mai mult, iar locurile de parcare par să nu fie niciodată suficiente. Așadar, mi-am propus să fac un sistem de parcare automatizat și inteligent, care să fie ușor de utilizat și optim.
- Acest sistem este util pentru noi deoarece aduce beneficii în ceea ce privește gestionarea locurilor de parcare și a spațiului disponibil pentru acestea, este ușor de folosit și automatizat, aplică concepte și tehnologii moderne pentru a rezolva o problemă importantă a oamenilor și pentru a îmbunătăți experiența utilizatorilor.

Descriere generală

Schemă bloc

- Senzorii de obstacole IR detectează trecerea unei mașini și se vor folosi întreruperi pentru a permite programului să răspundă imediat. Aceștia se află de o parte și de alta a barierei. Bariera este controlată de un servomotor, iar aceasta se deschide când unul din senzori detectează mișcare.
- Sistemul elevator este controlat din butoane. Acesta urcă și coboară cu ajutorul unui servomotor. LED-ul are diferite intensități ale luminii în funcție de nivelul unde se află liftul: superior sau inferior (PWM). Cu ajutorul display-ului cu I2C afișez activitatea curentă a parcării: înălțimea la care se află liftul (calculată aproximativ cu ajutorul senzorului ultrasonic) sau câte mașini au intrat în parcare.

- Ținte de performanță: timpul de reacție al barierei (aceasta se va ridica rapid, în mai puțin de o secundă de la detectarea mașinii), precizia senzorilor de obstacole (mașinile în miniatură sunt detectate la o distanță de aproximativ 5 cm de barieră), acuratețea afișării informațiilor.

Hardware Design

Componente:

- 1 X Arduino Uno
- 2 X Senzor IR
- 1 X Senzor ultrasonic
- 1 X Servomotor de rotație poziționalã
- 1 X Servomotor de rotație continuă
- 1 X Display LCD cu I2C
- 1 X Breadboard
- 2 X Butoane
- 1 X LED
- Fire de legătură + Rezistențe

Schemă electrică:

Software Design

- mediu de dezvoltare Arduino IDE
- concepte folosite: GPIO, întreruperi, timere, PWM, I2C
- biblioteci utilizate: Servo.h (pentru a controla cele două servomotoare), Wire.h (pentru comunicarea I2C), LiquidCrystal_I2C.h (pentru display), PinChangeInterrupt.h (pentru a crea întreruperi pe pinii care nu suportă întreruperi hardware în mod nativ

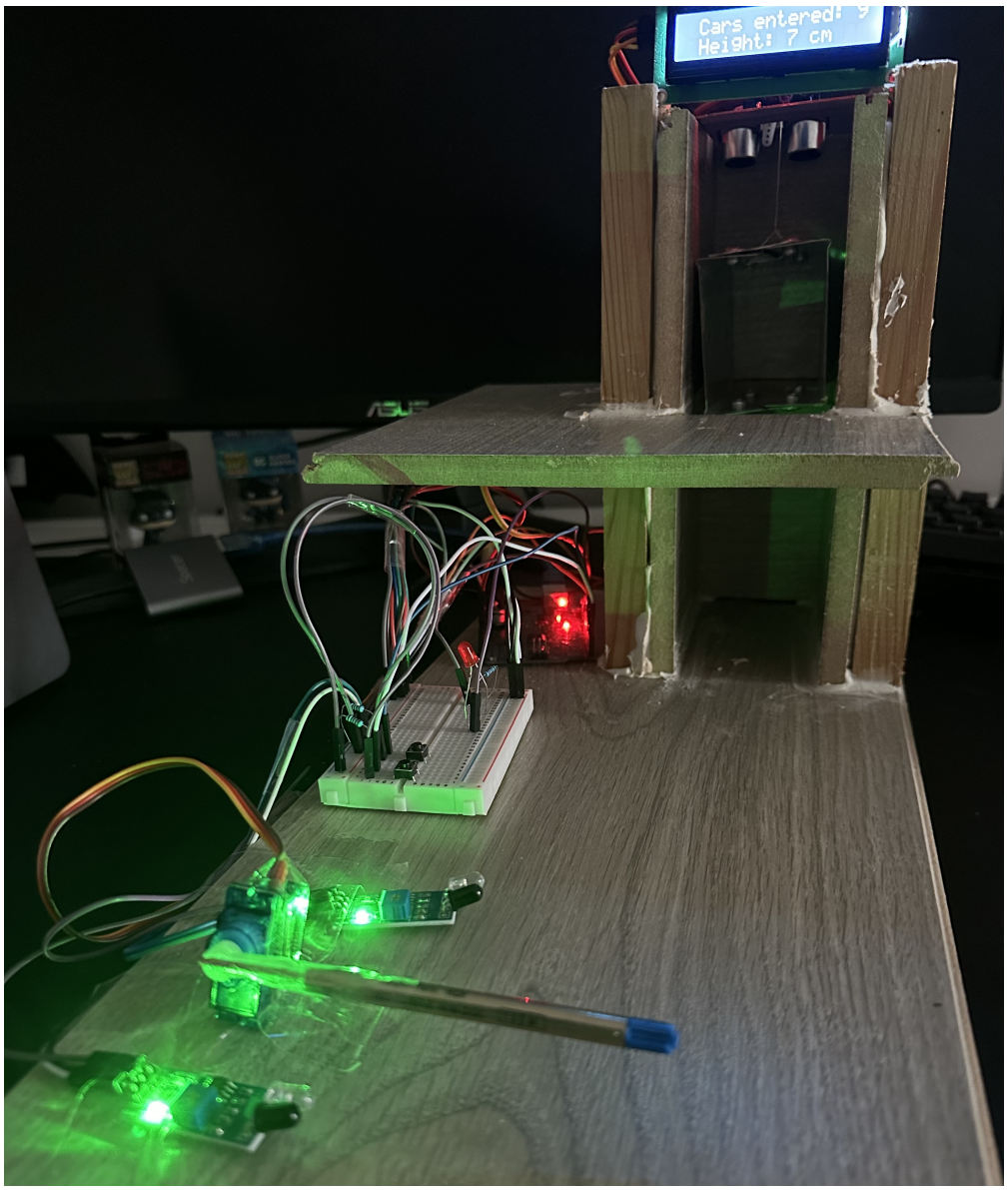
Funcții implementate:

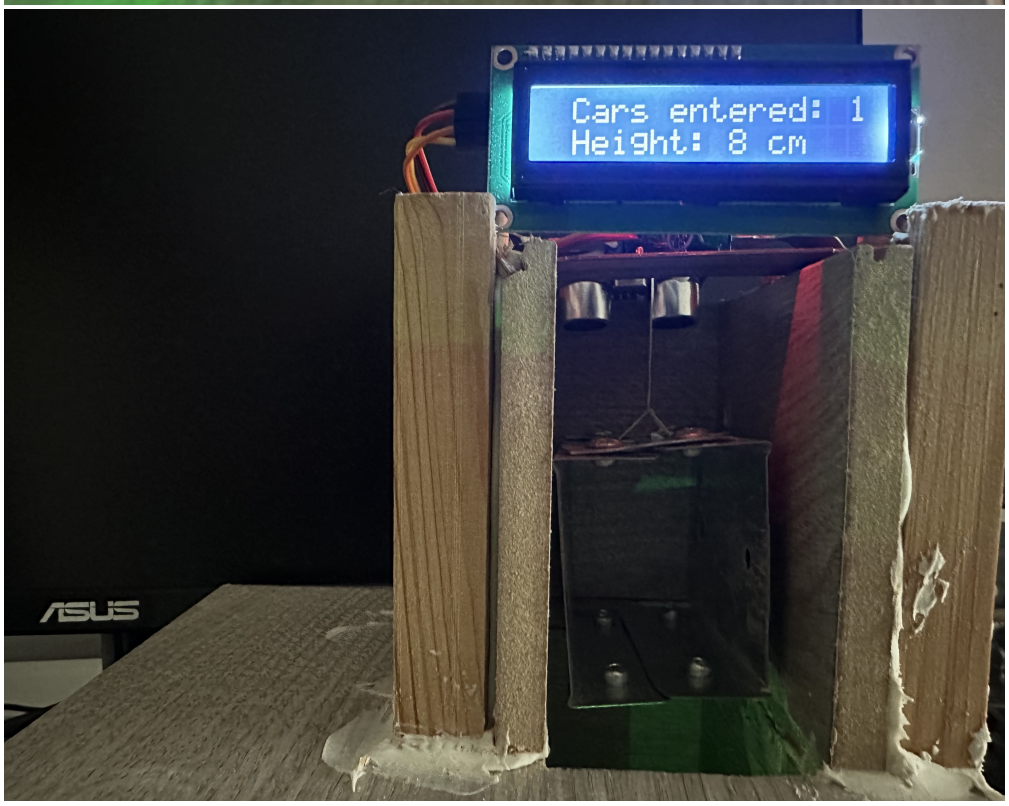
- setup(): configurează pinii de intrare și de ieșire, atașează întreruperi butoanelor pentru a detecta rapid apăsările și senzorilor IR pentru a detecta schimbările, atașează cele două servomotoare și le setează unghiul, inițializează LCD-ul. Folosesc conceptul de GPIO prin interacțiunea cu aceste componente electronice de input / output.
- button1ISR() și button2ISR(): setează flagurile corespunzătoare celor două butoane atunci când sunt

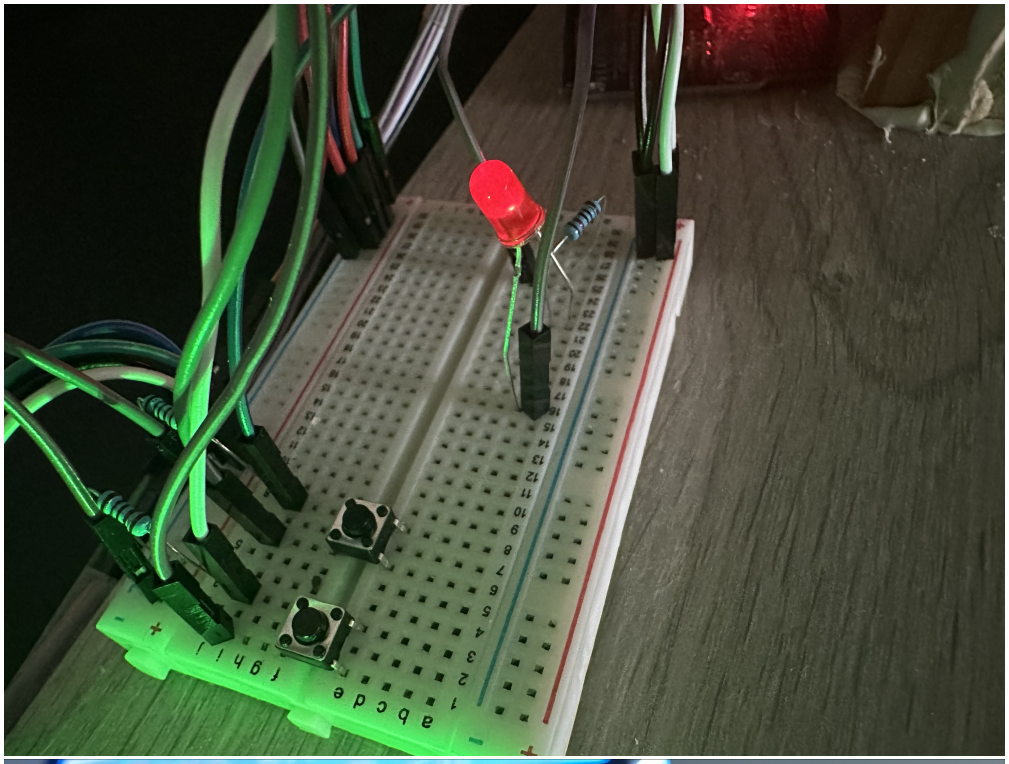
apăsate.

- irSensor1ISR() și irSensor2ISR() sunt rutine de întrerupere care detectează schimbarea senzorilor. Dacă senzorii nu detectează niciun vehicul, bariera va fi închisă după un timp de două secunde (am folosit timere — funcția millis()), pentru a lăsa suficient timp trecerii vehiculului. Dacă bariera detectează o mașină, atunci se deschide. De asemenea se ține incrementează / decrementează numărul de mașini care se află în parcare la momentul curent.
- measureDistance(int trigPin, int echoPin) măsoară distanța cu ajutorul senzorului ultrasonic. Trimite un puls de 10 ms pe pinul de trigger și măsoară durata de timp până la ecou. Calculează distanța în cm și o returnează.
- openBarrier() deschide bariera (90 grade) și closeBarrier() o închide (0 grade).
- increaseLedIntensity() crește treptat intensitatea folosind PWM, iar decreaseLedIntensity() o scade treptat prin aceeași metodă.
- moveElevatorUp() și moveElevatorDown() controlează mișcarea liftului (sus / jos). Pentru urcare, liftul funcționează până când senzorul ultrasonic măsoară o distanță de 3 cm între tavanul încăperii cu elevatorul și lift. Pentru coborâre, acesta se mișcă până când distanța crește peste 13.5 cm. Este calculată înălțimea curentă la care se află liftul și este afișată pe display-ul cu I2C. Intensitatea led-ului este maximă atunci când liftul este la parter (indicând etajul curent) și minimă atunci când liftul este la etajul superior.
- loop() verifică dacă a trecut o secundă de când a trecut ultima mașină și închide bariera. Crește contorul mașinilor, actualizează LCD-ul, verifică să nu se fi atins numărul maxim de mașini din parcare.

Rezultate Obţinute


Cars entered: 9
Height: 7 cm


Cars entered: 1
Height: 8 cm
ASUS



Concluzii

A fost un proiect interesant deoarece am învățat să construiesc un proiect hardware complex utilizând platforma Arduino. Am învățat să utilizez și să integrez diferite tipuri de senzori (IR, ultrasonic) în proiectul meu, să colectez și procesez date, să controlez servomotoare de mai multe tipuri, mi-am îmbunătățit cunoștințele despre întreruperi, PWM, configurarea display-urilor LCD.

Download

Implementarea software a proiectului se află pe GitHub:

https://github.com/DragosSerban/Proiect-PM/blob/main/proiect-pm.ino

Jurnal

Grafic Gant pentru descrierea sarcinilor

Bibliografie/Resurse

Utilizare display LCD și LiquidCrystal_I2C.h:

https://youtu.be/CvqHkXeXN3M?si=i-s5WMKo-jRrLlJO

Estimare distanță cu ajutorul senzorului ultrasonic:

https://www.geeksforgeeks.org/distance-measurement-using-ultrasonic-sensor-and-arduino/

Export to PDF