

Mini Barrier

Autor:  Frățiman Bogdan-Gabriel 

Grupa: 333CD

Introducere

Mini Barrier este un sistem automat de control al accesului, care detectează vehiculele care se apropie și acționează o barieră cu ajutorul unui servomotor. Utilizând un senzor ultrasonic, sistemul estimează distanța până la vehicul și decide automat dacă trebuie ridicată bariera. Pentru a permite și intervenția umană există și un buton care acționează manual bariera. Pe lângă acestea, un ecran LCD afișează distanța până la vehicul și contorizează numărul total de vehicule care au trecut prin sistem.

Scopul proiectului este de a simula un sistem inteligent de control al accesului vehiculelor similar celor întâlnite în parcuri, complexe rezidențiale sau centre logistice. Proiectul urmărește automatizarea unui proces des întâlnit, oferind date statistice în timp real, utile pentru gestionarea traficului prin sistem.

Ideea proiectului a pornit de la observarea sistemelor moderne de acces auto și dorința de a aduce un plus de eficiență prin reducerea intervenției umane, păstrând o evidență a vehiculelor, dar și siguranța sistemului și a șoferilor prin senzorul ultrasonic.

Acest sistem este util deoarece oferă o soluție pentru automatizarea accesului auto în spații precum parcuri private, curți, zone rezidențiale sau industriale. Prin automatizarea procesului de ridicare a barierei, se economisește timp, se reduc costurile cu personalul și se minimizează riscurile generate de erori umane. Ecranul LCD cu distanța și contorul de vehicule oferă un plus de transparență și control asupra traficului din zona respectivă.

Descriere generală

Mini Barrier este compus din următoarele module principale care interacționează între ele pentru a asigura buna funcționare a sistemului:



Fluxul de funcționare

1. Detecția unui vehicul: **Senzorul ultrasonic** măsoară continuu distanța și transmite date către

Arduino.

2. Procesarea logică: Dacă distanța este sub pragul ales, **Arduino** activează **servomotrul** pentru ridicarea barierei și modifică culoarea **LED-ului** corespunzător.
3. Afișare informații:
 - Pe **LCD** sunt afișate distanța până la vehicul, iar după ce acesta a trecut se incrementează numărul actual de vehicule
 - Pe **interfața serială** apare un log complet al activității.
4. Control manual: **Butonul** permite ridicarea/coborârea manuală a barierei.

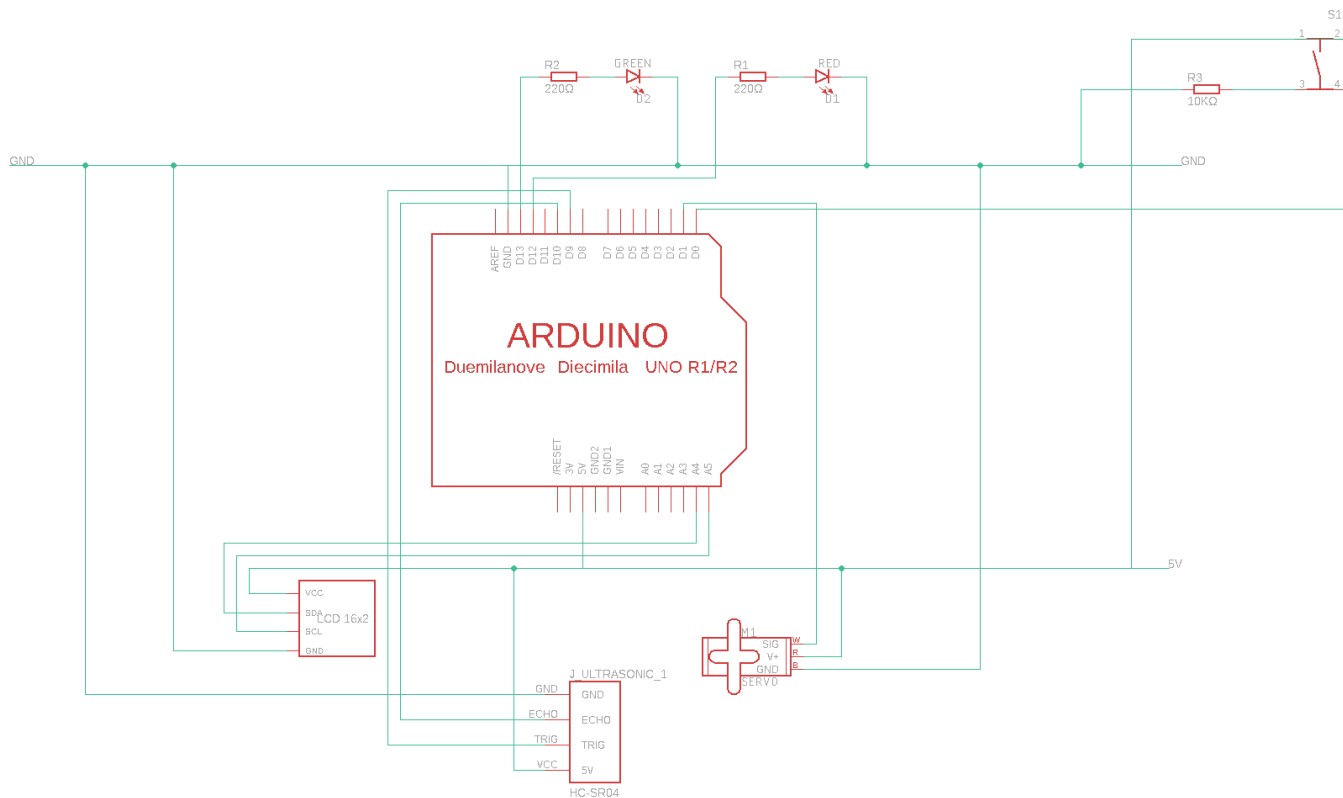
Hardware Design

✘ Listă componente: ✘

1. Placa Arduino UNO R3
2. Senzor ultrasonic HC-SR04
3. Servomotor SG90
4. LCD 16×2 cu adaptor I2C
5. Buton tactil
6. LED-uri
7. Rezistențe (220Ω, 10kΩ)
8. Breadboard
9. Cabluri jumper



Schema electrică



Software Design

Mediu de dezvoltare: Arduino IDE

Librării și surse 3rd-party: LiquidCrystal_I2C.h - pentru interfața I2C a LCD-ului

Algoritmi și structuri: Structura sistemului este asemănătoare unui state machine cu 4 stări: IDLE, CAR_IN, CAR_OUT și FULL, determinate de distanța măsurată și starea butonului, tratat ca întrerupere externă. Controlul servomotorului se face prin generarea unui semnal PWM folosind Timer1, configurat în mod fast PWM cu TOP = ICR1.

Surse și funcții implementate:

- setup() - Inițializează toate perifericele și configurează Timer1 în mod Fast PWM (WGM13:0 = 14) cu TOP = ICR1 = 40000, pentru a genera un semnal PWM de 50 Hz pe pinul OC1A (pin 9), cu un prescaler de 8, potrivit pentru servomotorul SG90:

```
TCCR1A = (1 << WGM11);
TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS11);

ICR1 = 40000;
TCCR1A |= (1 << COM1A1);
```

- setServoAngle(int angle) - Primește un unghi între 0° și 180°, îl convertește într-un impuls PWM între 750 și 2300 μs, apoi setează valoarea în OCR1A:

```
int pulseWidth = map(angle, 0, 180, 750, 2300);
OCR1A = pulseWidth * 2;
```

- `handleButtonPress()` - Activează rutina de întrerupere pentru buton setând flag-ul `buttonPressed`.

```
attachInterrupt(digitalPinToInterrupt(buttonPin), handleButtonPress, RISING);
```

- `loop()` - Bucloa principală a programului. Se ocupă de:
 - măsurarea distanței cu senzorul HC-SR04

```
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

long duration = pulseIn(echoPin, HIGH);
float distance = duration * 0.034 / 2;
```

- actualizarea ecranului LCD
- verificarea stării butonului
- acționarea barierei prin apeluri la `setServoAngle(...)`

Rezultate Obținute

Link Youtube:

<https://youtube.com/shorts/j4JR8kcPbFc?feature=share>

Concluzii

Proiectul **Mini Barrier** integrează mai multe componente hardware într-un sistem funcțional, de la controlul unui servomotor fără a folosi librăria `Servo.h`, prin configurarea directă a Timer-ului 1 în mod Fast PWM, până la tratarea butonului prin întrerupere externă și afișarea informațiilor pe un ecran LCD folosind protocolul I2C. Implementarea a demonstrat o aplicare practică a noțiunilor de bază din programarea microcontrolerelor, precum și modul în care acestea pot fi combinate într-o aplicație reală și interactivă.

Jurnal

M1 - Documentație - Am început redactarea paginii de OCW prin completarea următoarelor secțiuni: descriere, introducere, schema bloc și lista/descrierea componentelor hardware.

M2 - Hardware - Am adăgat schema electrică pe pagina de wiki și am realizat montajul parțial al proiectului atât hardware cât și software: LED-uri, buton, LCD.

M3 - Software - Am completat montajul și după ce m-am asigurat că toate componentele funcționează și logica state machine-ului este corectă am aplicat cunoștințele din laboratoare: Timer1 cu Fast PWM pentru servo, întreruperi pentru buton, I2C pentru LCD.

Bibliografie/Resurse

Repo Github: <https://github.com/bogdiw/mini-barrier>

Resurse utilizate:

<https://forum.arduino.cc/t/controlling-a-single-servo-using-pwm-and-timer-1/491621>

<https://www.youtube.com/watch?v=aQy3DGSIGm4>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/iotelea/bogdan.fratiman>



Last update: **2025/05/26 10:08**