

Smart Management Barrier Parking

Introducere

Proiectul consta intr-un sistem inteligent de management al accesului intr-o parcare, capabil sa valideze identitatea utilizatorilor, sa controleze o bariera fizica si sa tina evidenta locurilor disponibile in timp real. Sistemul interactioneaza atat cu lumea digitala (baze de date de carduri, afisaje), cat si cu cea fizica (actionarea mecanica a barierei si detectarea vehiculelor).

Scopul principal este realizarea unui sistem integrat care sa demonstreze lucrul simultan cu multiple protocoale de comunicatie (SPI, I2C) si periferice hardware interne (Timere, generare PWM).

Totul a plecat de la necesitatea optimizarii fluxului de masini intr-un spatiu inchis. Am pornit de la ideea unui sistem autonom care nu doar permite accesul oricum, ci urmareste fizic trecerea masinii pentru a evita erorile de contorizare.

Pentru mine, utilitatea este didactica si practica: invat sa stapanesc interactiunea dintre senzori asincroni (RFID, Ultrasonic) si actuatori (Servomotor) folosind o arhitectura de tip Finite State Machine (FSM), eliminand total functiile blocante de tip delay(). Pentru un utilizator real, sistemul ofera un feedback clar (pe ecranul LCD) si un acces fluid.

Laboratoarele folosite au fost 3-timere, PWM, 5-spi si 6-i2c

Descriere generala

Arhitectura software a sistemului este concepută sub forma unui automat cu stări finite (Finite State Machine), implementat în limbaj C la nivel de regiștri pe microcontrolerul ATmega328P. Sistemul este construit în jurul unei bucle principale non-blocking, care orchestrează comunicația cu perifericele, asigurând un management fluid al accesului auto și al resurselor parcării.

Fluxul operațional este împărțit în patru module logice principale:

1. Autentificare și Gestiunea Accesului (Comunicație SPI) Sistemul interoghează continuu senzorul RFID RC522 prin intermediul protocolului de comunicație de mare viteză SPI (Serial Peripheral Interface).

- **Validarea Identității:** La detectarea unui tag RFID (card sau breloc), microcontrolerul extrage identificatorul unic (UID) pe 4 octeți. Acest UID este comparat cu o structură de date internă ce conține utilizatorii autorizați.
- **Logica de Capacitate:** Chiar dacă un card este recunoscut ca fiind valid, accesul este condiționat de variabila globală a locurilor de parcare. Sistemul verifică preventiv dacă există locuri disponibile, refuzând accesul (și afișând un mesaj corespunzător pe ecran) în cazul în care parcare este la capacitate maximă.

2. Interfață și Feedback Vizual (Comunicație I2C) Feedback-ul către utilizator este asigurat de un ecran LCD 1602, controlat eficient prin protocolul I2C pentru a minimiza numărul de pini utilizați (se folosesc doar SDA și SCL).

- **Actualizare Dinamică:** Microcontrolerul actualizează interfața grafică în timp real, gestionând tranzițiile de stare: de la mesajul implicit de așteptare („Locuri Libere: X) , la mesaje de validare (, ,Acces Permis), sau mesaje de eroare („Card Invalid / , ,Parcare Ocupată).

3. Controlul Mecanic al Barierei (Timer Hardware și PWM) Pentru acționarea fizică a barierei (Servomotorul SG90), sistemul evită funcțiile de blocare a codului (precum funcțiile de delay), bazându-se pe resursele hardware interne ale microcontrolerului.

- **Generare Semnal Precis:** Este utilizat Timer-ul 1 pe 16-biți în modul Fast PWM. Prin setarea regiștrilor (ex: ICR1 și OCR1A), microcontrolerul generează un semnal PWM cu o frecvență fixă de 50Hz, modificând doar factorul de umplere (Duty Cycle) pentru a ridica bariera (poziția de 90 grade) la momentul validării accesului.

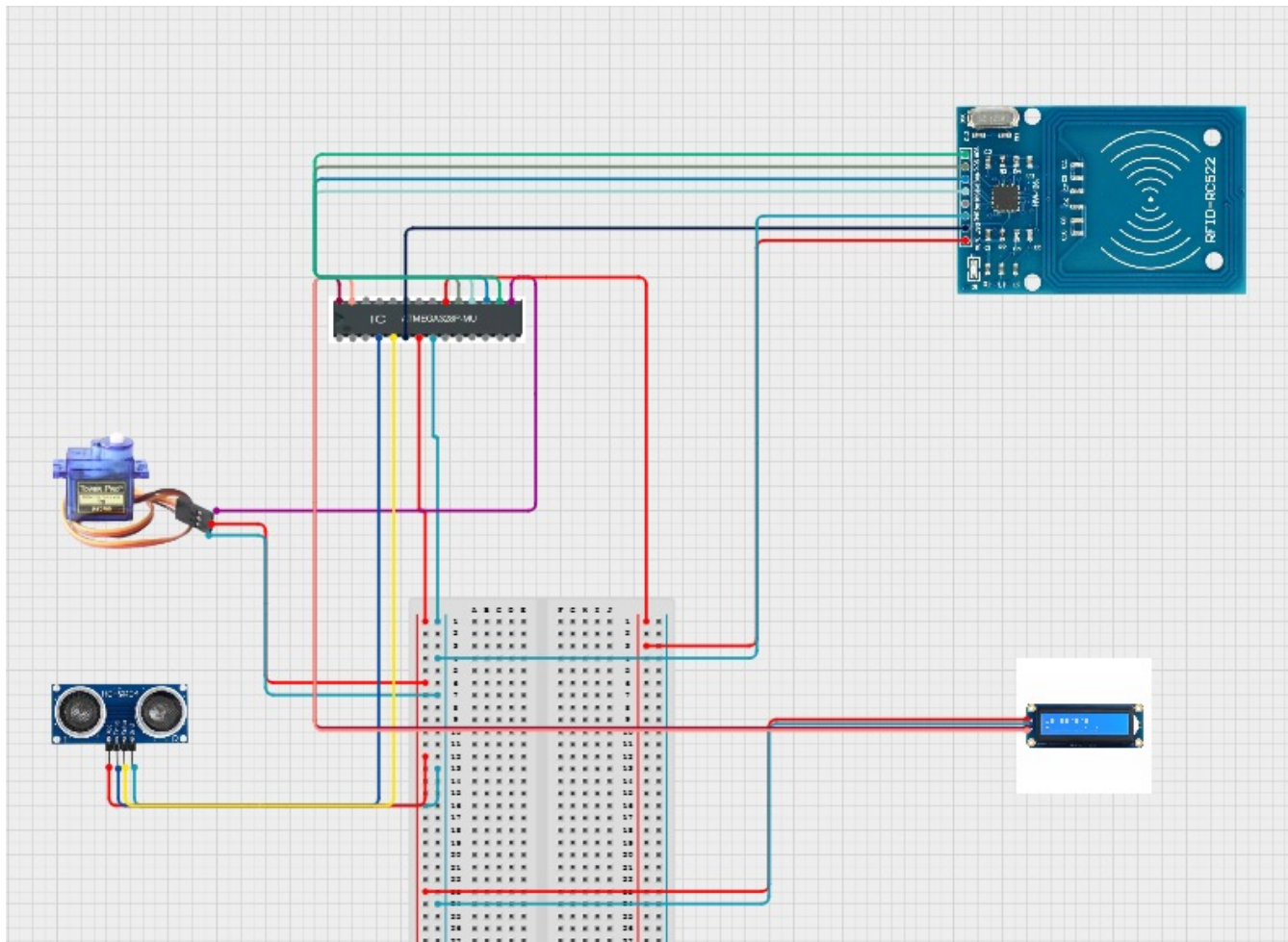
4. Telemetrie și Siguranța Tranzitului (Senzor Ultrasonic) Odată bariera ridicată, controlul logic este delegat modulului de siguranță gestionat de senzorul ultrasonic HC-SR04, pentru a preveni coborârea accidentală a brațului peste autovehicul.

- **Măsurarea Timpului de Zbor (ToF):** Microcontrolerul trimite un impuls scurt (Trigger) de 10 microsecunde și utilizează funcții de temporizare pentru a măsura durata impulsului de întoarcere (Echo).
- **Conversie și Decizie:** Distanța este calculată matematic pe baza vitezei sunetului în aer.
- **Logica de Tranzit:** Sistemul monitorizează continuu distanța. Când distanța scade sub un anumit prag, sistemul detectează mașina sub barieră. Doar în momentul în care distanța revine la valoarea inițială (confirmând că mașina a eliberat complet zona de tranzit), microcontrolerul comandă închiderea barierei și decrementează contorul locurilor de parcare, actualizând simultan afișajul LCD.



Hardware Design

Piesa	Link	Tip Utilizare
ATmega328P-Xplained Mini	ATmega328P-Xplained Mini	Baza dezvoltare
Modul RFID RC522	Modul RFID RC522	SPI
Display LCD 1602 cu convertor I2C	Modul interfață I2C pentru LCD	I2C
Servomotor SG90	Servomotor SG90	PWM (Timer1)
Senzor Ultrasonic HC-SR04	Senzor Ultrasonic HC-SR04	GPIO, Input Capture
Breadboard si fire DuPont	Breadboard si fire DuPont	Conexiuni hardware



Componentă	Pin componentă	Pin ATmega328P Xplained Mini	Explicație
HC-SR04	VCC	5V	Modulul ultrasonic este alimentat la 5V de la placă.
HC-SR04	GND	GND	Toate componentele trebuie să aibă masă comună.
HC-SR04	Trig (Trigger)	PD2	Pin digital configurat ca ieșire pentru generarea impulsului de declanșare (10μs).
HC-SR04	Echo	PD3	Pin digital configurat ca intrare pentru citirea ecoului (durata sa reprezintă distanța).
Servomotor SG90	VCC (Roșu)	5V	Alimentarea motorului (se recomandă condensator tampon pe breadboard dacă există fluctuații).
Servomotor SG90	GND (Maro/Negru)	GND	Masa comună.
Servomotor SG90	Semnal (Portocaliu)	PB1 / OC1A	Pin cu capacitate hardware PWM (Timer 1) pentru controlul unghiului brațului barierei.
LCD 1602 I2C	VCC	5V	Display-ul și modulul convertor sunt alimentate la 5V de la placă.
LCD 1602 I2C	GND	GND	Masa comună.
LCD 1602 I2C	SDA	PC4 / SDA	Pinul hardware standard pentru linia de date a magistralei I2C.

LCD 1602 I2C	SCL	PC5 / SCL	Pinul hardware standard pentru linia de ceas (clock) a magistralei I2C.
RFID RC522	VCC	3.3V	Atenție: Modulul funcționează STRICT la 3.3V. Alimentarea la 5V îl poate deteriora ireversibil.
RFID RC522	GND	GND	Masa comună.
RFID RC522	RST	PD4	Pin digital configurat ca ieșire pentru a controla resetarea hard a modulului.
RFID RC522	MISO	PB4 / MISO	Pinul standard hardware SPI pentru recepția datelor (Master In Slave Out).
RFID RC522	MOSI	PB3 / MOSI	Pinul standard hardware SPI pentru transmisia datelor (Master Out Slave In).
RFID RC522	SCK	PB5 / SCK	Pinul standard hardware SPI pentru generarea semnalului de ceas (Serial Clock).
RFID RC522	SS (SDA)	PB2 / SS	Pin configurat ca ieșire pentru a selecta modulul pe magistrala SPI (Slave Select).
RFID RC522	IRQ	Neconectat	Pin opțional pentru întreruperi, lăsat liber deoarece sistemul interoghează modulul direct din cod (polling).

Software Design

In stadiul actual, partea software a proiectului este functionala si integreaza toate componentele principale ale sistemului.

Functionalitatile implementate pana acum sunt:

- citirea UID-ului de pe cardurile RFID prin modulul RC522
- autorizarea cardurilor pe baza unei liste de UID-uri cunoscute, salvate in firmware
- controlul barierei (servomotor) prin semnal PWM generat pe Timer1
- detectarea trecerii masinii prin senzorul ultrasonic HC-SR04
- afisarea starii sistemului pe display-ul LCD 16x2 prin I2C (expander PCF8574)
- numararea locurilor libere si actualizarea automata dupa fiecare intrare confirmata
- tratarea cazului in care parcarea este plina, fara a ridica bariera
- tratarea cazului de timeout, in care cardul a fost scanat dar masina nu a trecut
- trimiterea de mesaje de debug prin UART pentru fiecare eveniment

La pornire, sistemul initializeaza toate perifericele si afiseaza pe LCD numarul de locuri libere din totalul de 10. Sistemul intra apoi intr-o bucla in care asteapta scanarea unui card. Dupa ce un card este detectat si validat, bariera se ridica, iar sistemul incepe sa monitorizeze senzorul ultrasonic pentru a confirma trecerea masinii inainte de a inchide bariera la loc si a decrementa contorul.

Structura codului

Codul este impartit in module clare, fiecare cu rol bine definit:

- initializare periferice: UART, I2C, SPI, RC522, servo (Timer1), HC-SR04, LCD
- modul SPI: configurare master, transfer pe 8 biti, control SS si RST pentru RC522
- modul RC522: scriere/citire registri, control antena, REQUEST, anticolizie si citire UID dubla pentru validare
- modul I2C: TWI master la ~100 kHz, doar operatii de scriere (LCD-ul nu necesita citire)
- modul LCD: driver HD44780 in mod 4-bit prin expander PCF8574, functii pentru clear, set_cursor, write_char/string/num
- modul display: functii de nivel inalt pentru ecranele recurente (idle si parcare plina)
- modul servo: PWM pe Timer1 (Fast PWM, ICR1 ca TOP), control pozitie inchis/deschis cu pre-pas pentru a forta miscarea
- modul sonar: trigger pe PD2, masurare durata ecou pe PD3, conversie in cm prin impartire la 58
- modul cards: lista statica de UID-uri autorizate si functia card_authorized() bazata pe memcomp
- modul UART: configurare 9600 baud 8N1 si functii pentru text, numere zecimale, hex si UID
- logica principala: main.cpp orchestreaza toate modulele intr-o masina de stari implicita

Interactiunea dintre functionalitati

Fluxul principal al aplicatiei este urmatorul:

- utilizatorul porneste sistemul
- perifericele sunt initializate si pe LCD apare mesajul "Scanati cartela" impreuna cu numarul de locuri libere
- bariera este coborata in pozitia inchisa
- sistemul intra in bucla principala si asteapta un card RFID
- cand un card este detectat, RC522 returneaza UID-ul prin SPI
- sistemul verifica daca UID-ul este in lista de carduri autorizate
- daca UID-ul nu este autorizat, LCD afiseaza "Card respins" si revine la idle
- daca UID-ul este autorizat dar parcare este plina, LCD afiseaza "PARCARE PLINA" si revine la idle
- daca UID-ul este autorizat si exista locuri libere, servoul ridica bariera
- sistemul monitorizeaza HC-SR04 pentru a detecta trecerea masinii
- dupa ce masina trece, bariera coboara, contorul de locuri libere este decrementat, iar LCD-ul afiseaza "Loc ocupat"
- daca masina nu este detectata in intervalul de timp permis, bariera coboara fara a decrementa contorul, iar LCD-ul afiseaza "Timeout!"
- sistemul revine la starea idle si asteapta urmatorul card

Prin aceasta logica, sistemul evita sa scada locurile libere daca masina nu trece efectiv prin dreptul senzorului, si nu ridica bariera in fata cardurilor neautorizate sau cand parcare este deja plina.

Detectia trecerii masinii

Pentru a confirma ca masina a trecut efectiv prin dreptul barierei am folosit senzorul ultrasonic HC-SR04. Algoritmul nu se bazeaza pe o singura citire, ci pe doua etape consecutive de filtrare:

- etapa 1 - detectia prezentei: sistemul polleaza senzorul la fiecare 100 ms; daca 3 citiri consecutive returneaza o distanta mai mica de 30 cm, se considera ca masina este sub senzor
- etapa 2 - detectia plecarii: dupa ce masina a fost detectata, sistemul continua sa polleze; daca 5

citiri consecutive returneaza o distanta mai mare sau egala cu 30 cm, se considera ca masina a parasit zona

Numarul de citiri consecutive necesare are rol de debounce si elimina citirile false cauzate de zgomotul ultrasonic sau de obstacole tranzitorii. Daca etapa 1 nu se incheie in 300 de cicluri (~30 secunde), tranzactia este considerata timeout si contorul nu este decrementat.

Controlul barierei

Bariera este actionata de un servomotor cu rotatie continua (CR), controlat prin PWM la 50 Hz generat de Timer1 in mod Fast PWM cu ICR1 = 40000 (TOP) si prescaler /8. Pentru acest tip de servomotor, latimea pulsului setata prin OCR1A nu dicteaza o pozitie fixa a bratului, ci directia si viteza de rotatie a motorului:

- stare de STOP (rotor oprit): OCR1A = 3000 (~1.5 ms)
- miscare sens deschidere: OCR1A = 2400 (~1.2 ms)
- miscare sens inchidere: OCR1A = 3600 (~1.8 ms)

Pentru a asigura o cursa corecta a barierei, functia `servo_set_position()` activeaza miscarea in directia dorita (deschis sau inchis) pentru un timp prestabilit (`ROT_TIME_MS`, calibrat pentru unghiul de ~90 de grade). Dupa expirarea acestui delay generat de `_delay_ms()`, semnalul PWM revine automat la valoarea de STOP (3000) pentru a frana si opri rotorul in noua pozitie. Aceasta calibrare inlocuieste controlul clasic in bucla inchisa al servomotoarelor standard.

Validarea functionalitatilor

Am testat fiecare componenta separat inainte de integrarea finala:

- RC522 a fost testat citind registrul de versiune (0x37) la pornire si trimitand valoarea prin UART pentru a confirma comunicarea SPI
- LCD-ul a fost testat initial cu un mesaj fix ("Sistem parcare / Initializare..."), apoi cu cifre si stringuri dinamice (contorul de locuri)
- servomotorul a fost testat separat alternand pozitiile `SERVO_INCHIS` si `SERVO_DESCHIS` pentru a regla valorile OCR1A
- HC-SR04 a fost testat trimitand distanta masurata prin UART si trecand obiecte prin fata senzorului
- algoritmul de autorizare a fost testat cu carduri din lista (UID 0x85 0x64 0xF3 0x06 si 0xE6 0xD7 0x10 0x06) si cu carduri necunoscute, pentru a verifica ambele ramuri (acces permis / acces respins)
- fluxul complet a fost testat scanand card, ridicand bariera, simuland trecerea unei masini (mana in fata senzorului) si verificand decrementarea contorului

Pentru debug am folosit UART la 9600 baud, unde se afiseaza fiecare eveniment important: UID-ul citit, statusul autorizarii, momentul ridicarii barierei, detectia masinii, timeout-urile si starea contorului dupa fiecare tranzactie.

Optimizari realizate

O prima optimizare a fost dubla citire a UID-ului in `rc522_read_card()`. Prima citire se face, apoi dupa 50 ms se face a doua citire si se compara cu prima. UID-ul este acceptat doar daca cele doua citiri sunt identice. Aceasta optimizare elimina UID-urile gresite cauzate de coliziuni partiale sau de carduri scoase din campul antenei in timpul citirii.

A doua optimizare a fost filtrarea UID-urilor invalide (toate octetii `0x00` sau toti `0xFF`). Aceste valori apar uneori cand antena RC522 detecteaza zgomot electromagnetic, dar nu corespund unui card real, si rejectandu-le se evita interpretarea zgomotului ca un card neautorizat.

A treia optimizare a fost resetarea campului antenei dupa fiecare tranzactie prin `rc522_reset_field()`. Aceasta opreste antena pentru 100 ms si o porneste din nou, ceea ce forteaza cardul aflat in apropiere sa fie recitit ca o tranzactie noua, in loc sa fie tratat ca o citire repetata.

A patra optimizare a fost folosirea detectiei in doua etape la HC-SR04 (prezenta + plecare, fiecare cu praguri de citiri consecutive). Aceasta abordare evita atat alarmele false (un obiect care apare o singura citire), cat si decrementarea prematura a contorului (in care masina inca este sub senzor cand bariera se inchide).

Element de noutate

Elementul de noutate al proiectului este faptul ca sistemul nu se limiteaza la a ridica bariera la scanarea unui card, ci confirma fizic trecerea masinii inainte de a marca locul ca fiind ocupat. Sistemul diferentiaza intre trei situatii:

- cardul a fost scanat dar masina nu a trecut (timeout) - bariera se inchide fara a decrementa contorul
- cardul a fost scanat si masina a trecut - bariera se inchide si contorul scade cu 1
- cardul a fost scanat dar parcare este plina - bariera nu se ridica deloc

In plus, prin combinatia dintre verificarea dubla a UID-ului si filtrarea citirilor de zgomot pe antena RC522, sistemul evita sa accepte sau sa respinga in mod fals un card pe baza unei singure citiri instabile. Astfel, sistemul ofera o sincronizare reala intre evenimentul de autorizare si miscarea fizica a masinii, lucru care nu ar fi posibil cu un simplu cititor RFID conectat direct la actuator.

Rezultate Obtinute

[Link github](#)

[Link video demo](#)

Bibliografie/Resurse

Resurse Hardware:

- Datasheet ATmega328P: Documentatia oficiala Microchip pentru registrii TWI, SPI si Timere.
- Datasheet MFRC522: Documentatia NXP pentru harta registrilor modulului RFID.
- Datasheet HD44780: Setul de instructiuni pentru ecranul LCD.

Resurse Software:

- Laboratoarele PM UPB: sectiunile legate de SPI, PWM, ADC si I2C.
- Coduri de referinta GitHub pentru adaptarea MFRC522 in C.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/theodor.mitrofan>



Last update: **2026/05/25 10:03**