

Swiffer Autonom

Introducere

Proiectul consta in crearea unui robot "autonom" care urmareste un algoritm de tip zig-zag. Acesta poate fi oprit, pornit sau resetat prin intermediul modului Bluetooth. Pe viitor se poate adapta la a se conecta la un server ML pentru a invata camera autonom, iar Swiffer-ul poate fi schimbat cu un modul de aspirator.

Descriere generală



Descrierea sumara a modulelor si a modului de interactiune

1. Descrierea modulelor hardware:

- **Unitatea de procesare (ATmega328P):** Reprezinta "creierul" sistemului, fiind responsabil cu rularea algoritmului principal si coordonarea tuturor perifericelor.
- **Modulul de alimentare si putere (L298N & Divizor):** Gestioneaza energia. Regulatorul integrat asigura 5V stabili pentru logica sistemului, puntea H controleaza curentii mari pentru locomotie, iar divizorul de tensiune permite citirea nivelului bateriei in conditii de siguranta.
- **Modulul de perceptie (Senzori HC-SR04 si TCRT5000):** Culeg date din mediul fizic, masurand distanta pana la pereti/obstacole si detectand prezenta suprafetei de rulare.
- **Sistemul de locomotie (Motoare DC):** Executa miscarea fizica (directie si viteza) a sasiului.
- **Modulul de comunicatie (Bluetooth HC-05):** Asigura o legatura seriala wireless bidirectionala cu un terminal extern pentru monitorizare si debugging.

2. Modul de interactiune (Fluxul de functionare):

Sistemul interactioneaza printr-o bucla continua de tip **Achizitie date → Procesare → Actiune**. Pachetul de baterii alimenteaza intregul ansamblu. In timpul rularii, senzorii colecteaza date din mediu si le transmit catre microcontroler sub forma de impulsuri sau niveluri de tensiune. ATmega328P analizeaza aceste intrari si, pe baza algoritmului de decizie, trimite semnale de control (PWM si directie logica) catre driverul L298N. Driverul actioneaza ca un amplificator, cupland motoarele la curentul bateriei pentru a executa deplasarea. In paralel, microcontrolerul raporteaza constant starea sistemului (baterie, distante, decizii de viraj) prin intermediul modului Bluetooth.

Hardware Design

1. Listă de piese (BOM - Bill of Materials)

Pentru realizarea acestui proiect, a fost utilizată următoarea configurație hardware, axată pe un echilibru între eficiența energetică și acuratețea senzorilor:

- **Unitate de Control:** 1x Placă de dezvoltare ATmega328P Xplained Mini.
- **Sistem de Locomoție:**
 - 1x Șasiu robot 2WD (din plexiglas).
 - 2x Motoreductoare de curent continuu (Motoare TT, raport 1:48).
 - 2x Roți cauciucate + 1x Roată pivotantă (Caster wheel).
 - 1x Modul Driver Motoare L298N (cu regulator de tensiune LM7805 integrat).
- **Sistem de Percepție (Senzori):**
 - 3x Module ultrasonice HC-SR04 (amplasate frontal, stânga, dreapta).
 - 1x Senzor optic reflexiv infraroșu TCRT5000 (pentru detecția marginilor / "Swiffer").
- **Sistem de Comunicație:** 1x Modul Bluetooth HC-05.
- **Sistem de Alimentare:**
 - 1x Suport pentru 4 baterii tip AA.
 - 4x Baterii Alcaline AA 1.5V (Tensiune totală nominală: 6V).
- **Componente Pasive & Conectică:**
 - 1x Rezistor 4.7kΩ (Albastru-precizie: Galben-Violet-Negru-Marou).
 - 1x Rezistor 2.2kΩ (Albastru-precizie: Roșu-Roșu-Negru-Marou).
 - 1x Breadboard mini (170 puncte).
 - Set fire conexiune Dupont (Tată-Tată, Mamă-Tată).

2. Scheme Electrice și Maparea Pinilor

Pentru a vizualiza interconectarea componentelor, se va consulta schema electrică de ansamblu de mai jos. *(Notă tehnică: Alimentarea logicii de 5V a microcontrolerului se face prin pinul de +5V al modulului L298N, acesta coborând tensiunea de 6V a bateriei printr-un regulator integrat).*



Tabelul Conexiunilor Hardware (Pinout Map): Pentru interfațarea senzorilor cu perifericele interne ale ATmega328P s-a utilizat următoarea mapare:

Modul Extern	Pin Modul	Conexiune ATmega328P	Rol / Periferic Utilizat
Modul HC-05	TX	PD0 (RXD)	Comunicație Serială (USART)
Modul HC-05	RX	PD1 (TXD)	Comunicație Serială (USART)
L298N (Driver)	ENA	PD5	Control Viteză Motor Stânga (PWM Hardware)

L298N (Driver)	ENB	PD6	Control Viteză Motor Dreapta (PWM Hardware)
L298N (Driver)	IN1, IN2	PD4, PD7	Control Direcție Motor Stânga (GPIO)
L298N (Driver)	IN3, IN4	PB2, PB3	Control Direcție Motor Dreapta (GPIO)
Divizor Tensiune	Vout (Intersecție)	PC0 (A0)	Monitorizare nivel baterie (ADC)
TCRT5000	A0	PC1 (A1)	Detectie analogică margine / podea (ADC)
HC-SR04 (Față)	Trig, Echo	PD2, PD3	Măsurare distanță față (GPIO)
HC-SR04 (Stânga)	Trig, Echo	PC2 (A2), PC3 (A3)	Măsurare distanță stânga (ADC folosit ca GPIO)
HC-SR04 (Dreapta)	Trig, Echo	PC4 (A4), PC5 (A5)	Măsurare distanță dreapta (ADC folosit ca GPIO)

3. Diagrame de Semnal

Pentru interfațarea corectă a perifericelor s-au analizat următoarele semnale cheie:

A. Diagrama de semnal pentru senzorii HC-SR04 (Time-of-Flight) Microcontrolerul emite un impuls logic HIGH de 10μs pe pinul Trigger. Senzorul răspunde cu un semnal HIGH pe pinul Echo, a cărui durată este proporțională cu distanța. Această durată este măsurată hardware folosind întreruperile externe și un Timer.



B. Diagrama semnalului PWM pentru L298N Controlul locomotiei și aplicarea metodei de “Soft Start” se realizează prin modularea lățimii impulsurilor (PWM) aplicate pe pinii ENA și ENB ai driverului, dictând astfel cuplul și viteza motoarelor.



Software Design

(Nota: Aceasta sectiune prezinta o arhitectura de nivel inalt, detaliile urmand sa fie rafinate in etapa finala de implementare).

1. Mediu de dezvoltare

Pentru scrierea, compilarea si incarcarea firmware-ului, mediul de dezvoltare ales este **Arduino IDE**. Desi platforma hardware este un ATmega328P Xplained Mini, ecosistemul Arduino ofera un flux de lucru mult mai rapid (Rapid Prototyping) si o sintaxa mai accesibila pentru gestionarea pinilor (GPIO) si a functiilor de timp, comparativ cu programarea stricta pe registri in Microchip Studio.

2. Librarii si surse 3rd-party

Pentru a mentine un control strict asupra executiei si a minimiza overhead-ul, utilizarea librariilor externe va fi limitata la minimum necesar:

- Se va folosi exclusiv biblioteca standard **Arduino Core** (pentru functii de baza precum `millis()`, `analogRead()`, `analogWrite()`, `digitalWrite()`).
- Interfatarea cu senzorii ultrasonici si citirea nivelului bateriei se vor face prin metode standard, fara a importa librarii masive din surse terte.
- Optional, pentru modulul Bluetooth, se poate folosi libraria `SoftwareSerial.h` inclusa in pachetul standard Arduino.

3. Algoritmi si structuri de date planificate

Aplicatia va fi proiectata pe o arhitectura **non-blocanta** (fara utilizarea instructiunii `delay()`), esentiala pentru reactia in timp real a robotului la datele primite de la senzori.

- **Multitasking cooperativ:** Se va utiliza functia `millis()` pentru a implementa "software timere". Acest algoritm permite procesorului sa citeasca senzorii, sa raporteze date prin Bluetooth si sa ajusteze PWM-ul motoarelor (aproape) simultan.
- **Masina de Stari Finita (FSM):** Structura principala de control va fi un bloc switch-case ce defineste comportamentul robotului. Starile planificate includ: `MERGE_FATA`, `DETECTARE_PERETE`, `INTOARCERE_90_GRADE_ST/DR`, `ALINIERE_ZIG_ZAG`, `OPRIRE_URGENTA_MARGINI` si `LOW_BATTERY`.
- **Algoritm de acoperire:** Logica FSM va dicta un traseu de tip zig-zag. La detectarea unui obstacol frontal, robotul va verifica senzorii laterali pentru a alege directia corecta de viraj, alternand intoarcerile stanga-dreapta.

4. Surse si functii implementate (Etapa 3)

(Sectiunea va fi completata cu fragmente de cod in momentul finalizarii dezvoltarii. Principalele functii planificate sunt:)


- `void setup()` si `void loop()` - structura de baza Arduino.
- `void readSensors()` - responsabila de declansarea impulsurilor Trig si citirea duratei Echo, precum si citirea ADC pentru TCRT5000 si baterie.
- `void motorControl(int vitezaStanga, int vitezaDreapta)` - o functie de abstractizare ce transpune cerintele de miscare in semnale PWM si stari logice pentru driverul L298N.
- `void updateFSM()` - functia care proceseaza datele de la senzori si decide comutarea dintr-o stare in alta.

Rezultate Obținute

To be determined.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

==== Resurse Hardware (Datasheet-uri) ====

- **ATmega328P Xplained Mini** - User Guide-ul oficial Microchip (utilizat pentru maparea pinilor fizici ai placii si schemele de alimentare): [ATmega328P Xplained Mini User Guide \(PDF\)](#)
- **Microcontroler ATmega328P** - Datasheet complet (utilizat pentru intelegerea arhitecturii, registratorii interni, Timere, ADC, USART si intreruperi): [ATmega328P Datasheet \(PDF\)](#)
- **Driver Motoare L298N** - STMicroelectronics Datasheet (consultat pentru limitele de curent, caderea de tensiune pe puntea H si logica de control): [L298N Datasheet \(PDF\)](#)
- **Senzor Ultrasonic HC-SR04** - Specificatii tehnice (utilizat pentru extragerea diagramelor de semnal si a formulei de calcul a distantei bazate pe viteza sunetului): [HC-SR04 Datasheet \(PDF\)](#)
- **Modul Bluetooth HC-05** - Manual de utilizare (consultat pentru tensiunile de operare si pinii de comunicatie RX/TX): [HC-05 Datasheet \(PDF\)](#)
- **Senzor Infrarosu TCRT5000** - Vishay Datasheet (utilizat pentru curbele de reflectie si intelegerea circuitului emitor-receptor): [TCRT5000 Datasheet \(PDF\)](#)
- **Motoare DC cu Reductor (TT Motors 1:48)** - Specificatii generice pentru motoarele galbene de casiu (utilizate pentru a calcula curentul de stall necesar a fi suportat de L298N): [TT Motor Datasheet \(PDF\)](#)

===== Resurse Software =====

- **OCW UPB - Proiectare cu Microprocesoare:** Suportul teoretic de laborator pentru configurarea perifericelor hardware si notiuni de electronica: ocw.cs.pub.ro/courses/pm
- **Arduino Reference:** Documentatia oficiala pentru arhitectura de cod si functiile standard (millis, pini I/O) utilizate in sistemul non-blocant: [Arduino Language Reference](#)
- **WaveDrom:** Unealta utilizata pentru generarea diagramelor de semnal (timing diagrams) din sectiunea de Hardware Design: wavedrom.com

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2026/vlad.radulescu2901/rares.ciocia>



Last update: **2026/05/13 08:18**