

Willy - the Buddy Chaser

Introducere

„**Willy - the Buddy Chaser**” este o masinuta autonoma proiectata sa urmareasca un om in timp real, folosind senzori de detectie (cu infrarosu), un senzor ultrasonic si un microcontroler. Scopul principal al proiectului este dezvoltarea unui sistem hardware capabil sa identifice si sa urmareasca un obiect in miscare, integrand concepte esentiale din domeniul proiectarii cu microprocesoare.

Ideea a pornit din dorinta de a combina elemente de robotica cu o componenta interactiva și prietenoasa. Pe langa aspectele tehnice, proiectul are si un scop distractiv – utilizatorul se poate juca efectiv cu masinuta, fiind urmarit de aceasta intr-un mod similar cu un „*animal de companie robotic*”. Aceasta latura face ca proiectul sa fie mai atractiv si accesibil inclusiv pentru publicul larg.

Acest proiect poate reprezenta o baza pentru aplicatii educationale, jucarii inteligente sau chiar sisteme autonome de asistenta in viitor.

Descriere generală

Schema bloc



Hardware Design

Lista piese

Denumire piesa	Link achizitionare
Arduino UNO R3	Optimus digital
Motor Driver L298N	Optimus digital
Senzor Infrarosu de Obstacole	Optimus digital
Senzor Ultrasonic de Distanta HC-SR04+	Optimus digital
Motor cu reductor si roata	Optimus digital
Mini Breadboard Albastru	Optimus digital
LCD 1602	Optimus digital
Buzzer	Optimus digital

Suport 6 baterii AA	Optimus digital
Comutator KCD10-101	Optimus digital
Set Fire tata-tata	Optimus digital
Set fire mama-tata	Optimus digital

Schema Electrica



Schema Board



Software Design

Descrierea generala a aplicatiei (Firmware)

Aceasta aplicatie controleaza un robot mobil autonom („Willy”), dotat cu:

- Senzori ultrasonici pentru detectia obstacolelor.
- Senzori infrarosu (IR) pentru urmarirea unui obiect (de exemplu, o mana).
- Motoare DC pentru miscare si manevrare.
- Buzzer pentru semnalizare sonora.
- Ecran LCD I2C 16x2 pentru afisarea starii robotului.

Robotul functioneaza in mod **follow-me**: detecteaza si urmareste un obiect apropiat. Se opreste sau se retrage cand identifica un obstacol.

Mediu de dezvoltare

- **Platforma hardware**: Arduino UNO (ATmega328P)
- **Mediu de dezvoltare**: Arduino IDE 2.x
- **Limbaj de programare**: C++ (Arduino Sketch)
- **Placa suport**: Arduino UNO R3

Librarii si surse 3rd-party utilizate

Biblioteca	Descriere	Link
------------	-----------	------

`NewPing`	Permite masurarea precisa a distantei cu senzorul ultrasonic HC-SR04.	NewPing Library
`Wire`	Protocol I2C pentru comunicatia cu ecranul LCD.	Inclusa in Arduino IDE
`LiquidCrystal_I2C`	Suport pentru afisaj LCD 16x2 cu interfata I2C.	LiquidCrystal_I2C

Algoritmi si structuri implementate

Algoritmi principali

• Controlul miscarii robotului:

- Functia ``rotateMotor()`` permite controlul diferential al motoarelor, in functie de detectiile senzorilor IR sau ultrasonic.
- Viteza ajustabila: ``MAX_SPEED`` + ``ADJUST_SPEED`` (forward) sau diferentiere la viraje (``+/-``).

• Detectia obstacolului cu senzor ultrasonic:

- Cu biblioteca ``NewPing``, masuram distanta in centimetri.
- Daca distanta < 2 cm, se activeaza un semnal sonor (``buzzer``) si robotul se opreste, se retrage (da putin cu spatele), apoi se opreste complet.

• Urmărirea obiectului cu senzori IR:

- Daca ambii senzori IR detecteaza obiect (LOW), robotul merge inainte (``Chasing``).
- Daca doar un senzor detecteaza (LOW), robotul vireaza in directia respectiva (``Turning Left`` / ``Turning Right``).
- Daca nu detecteaza, se opreste (``Waiting...``).

• Afisarea starii pe LCD:

- Starea robotului (``Chasing``, ``Turning``, ``Obstacle``, ``Waiting``) este afisata impreuna cu caractere customizate (emoji, sageti).
- Bara de progres (``distance bar``) arata distanta fata de obstacol (linie pe randul 2 al LCD).

Structuri de date

• Variabile globale pentru:

- Viteze motoare (``currentRightSpeed``, ``currentLeftSpeed``).
- Stare obstacol (``check_close``).
- Ultimul caracter afisat pe ecran (``lastStatus`` - optional pentru optimizare).

• Caractere custom (``byte SmileyFaceChar[]``, ``byte StopChar[]``, ``byte leftArrowChar[]``, ``byte rightArrowChar[]``).

Etapa 3: Surse si functii implementate

Fisiere sursa

- ``main.ino`` – Codul principal (Arduino Sketch).

Functii implementate

Funcție	Rol
<code>`setup()`</code>	Initializeaza pinii, LCD-ul, motoarele, senzorii.
<code>`loop()`</code>	Logica principala a programului: detectie obstacole, urmarire IR, update LCD, buzzer.
<code>`rotateMotor()`</code>	Controleaza directia si viteza motoarelor in functie de parametrii.
<code>`displaySpeed()`</code> (nefolosit)	Afiseaza viteza pe LCD (optional, comentat).
<code>`lcd.createChar()`</code>	Definirea caracterelor custom pentru LCD (smiley, stop, sageti).

Rezultate Obținute

- In urma realizarii proiectului, am obtinut un sistem functional care controleaza o **masina autonoma** capabila sa efectueze urmatoarele actiuni:
 - **Urmarire:** Masina detecteaza un obiect (mana) in fata senzorilor IR si incepe sa-l urmareasca, ajustandu-si directia in functie de pozitia obiectului.
 - **Afisare pe LCD:** Mesajele afisate pe ecranul LCD reflecta starea curenta a masinii, precum:
 - Chasing (urmarire)
 - Turning Left (viraj la stanga)
 - Turning Right (viraj la dreapta)
 - Waiting... (asteptare)
 - OBSTACLE AHEAD! (obstacol detectat)
 - **Evitarea obstacolelor:**
 - Daca senzorul ultrasonic detecteaza un obstacol prea aproape (sub 2 cm), masina:
 - Opreste motoarele.
 - Afiseaza mesajul de avertizare pe LCD.
 - Emite un **bip** de avertizare prin buzzer.
 - Executa o miscare de **recul** (da putin cu spatele) pentru a se indeparta de obstacol.
- Sistemul integreaza mai multe concepte hardware si software:
 - Controlul motoarelor cu semnale **PWM**.
 - Senzori IR pentru detectarea obiectelor.
 - Senzor ultrasonic pentru masurarea distantei.
 - Buzzer pentru semnalizare sonora.
 - Ecran LCD pentru afisare in timp real a starii.

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

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/apredescu/cristian.deda>



Last update: **2025/05/31 20:42**