

Documentație Robot de urmărire

Proiect la

Sisteme cu microcontrollere și IOT industriale

Stanislav Semenenco

Alina Brînza

Mihail Zatić

Cristian Gurduza

Profesor:

Conf.dr.ing. Adrian Korodi

Timișoara

<2023>

Cuprins

1. [Introducere](#)
2. [Baze teoretice](#)
 - A. Diagrama bloc a sistemului
 - B. Detectarea obiectului
 - C. Senzorul Pixy CmuCam5
 - D. Arduino Uno
 - E. Driver L298N
3. [Implementare](#)
 - A. Structura mecanică
 - B. Partea electronică și hardware
 - C. Design Firmware
4. [Bibliografie](#)

1. Introducere

Utilizarea roboților este pe larg răspândită în industrie și este de mare folos pentru omenire, astfel roboții sunt utilizați pentru irigație, stingerea focului sau dezinfecția în spitale.

În acest proiect, robotul este controlat de o placă Arduino Uno cu microcontroller. Utilizarea acestei componente este considerată pentru simplitatea sa.

Placa Arduino are un microcontroler Atmega328 IC care este foarte relevant de utilizat în sisteme de automatizare simple și în timp real.

Senzorul Pixy Cmcam5 are capabilități de procesare a imaginii. Acest senzor este plasat în fața robotului și are un unghi de vizualizare larg. Acest senzor generează date despre dimensiunea obiectului și poziția acestuia pentru a fi transmise plăcii Arduino prin comunicație serială. Mișcarea robotului se face pe baza datelor reale de la acest senzor Pixy.

2. Baze teoretice

A. Diagrama bloc a sistemului

Senzorul Pixy este utilizat pentru a detecta dimensiunea și coordonatele obiectelor într-o imagine bidimensională.

Rezultatele detectării sunt trimise la placa cu microcontroler Arduino Uno pentru prelucrare. Poziția obiectului în imaginea bidimensională va determina direcția de mișcare a robotului, care este realizată de 2 motoare: drept și stâng. Controlul de viteză și activarea celor două motoare sunt efectuate de un Driver de comandă motoare L298N.

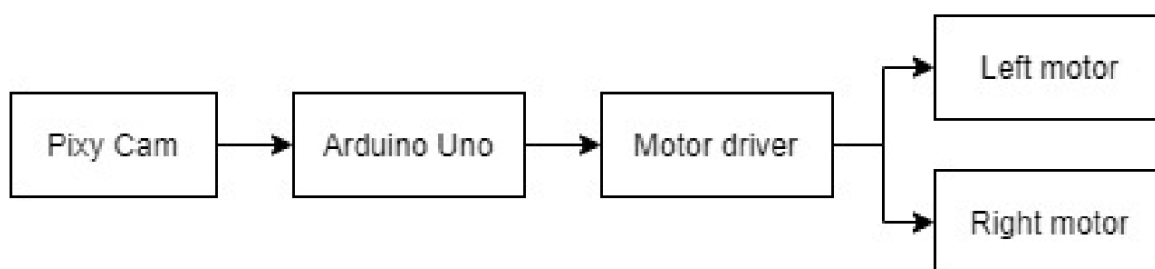


Fig. 1. Diagrama bloc a sistemului

Mișcările: înainte, înapoi, dreapta, stânga și staționar reglează poziția obiectului, astfel încât robotul să poată urmări poziția obiectelor în spațiu.

B. Detectarea obiectului

Detectarea obiectelor se face utilizând camera și are nevoie de unele configurări și setări. Camera prima dată are nevoie să se focalizeze pe obiect, iar pentru asta e necesară o iluminare corespunzătoare, deoarece camera e sensibilă la lumină. Apoi, imaginea unui obiect îl transformă într-un semnal video care este apoi digitizat și stocat în memoria microcontrolerului. Datele imaginii capturate trebuie calculate fiecare frame.

C. Senzorul Pixy CmuCam5

Procesorul dual core NXP LPC4330 încorporat în Camera Pixy CMUCam5 este locul unde are loc procesarea imaginii. Pixy Camera CMUCam5 este capabilă de procesare integrată a imaginii și poate urmări anumite culori în același timp. Obiectul folosit este hârtie colorată cu formă dreptunghiulară. Aplicația PixyMon poate fi folosită pentru a eticheta obiecte. Această cameră are un unghi de vizualizare de 75 de grade. Coordonatele obiectului determinate de cameră, vor fi baza pentru reglarea mișcării a 4 motoare.

D. Arduino Uno

Arduino Uno este un kit de dezvoltare bazat pe microcontroller-ul ATmega28. Acest modul este echipat cu diverse lucruri necesare ca microcontrollerul să funcționeze, e necesar doar să fie alimentat prin cablu USB sau de la o sursă externă de alimentare.

Placa Arduino Uno are 14 pini de intrare/ieșire digitali, 6 intrări analogice, un rezonator ceramic de 16 MHz, o conexiune USB, o mufă de intrare de alimentare, un antet ICSP și un buton de resetare.

E. Driver L298N

L298N este un driver de motor de mare putere pentru conducerea motoarelor de curent continuu (DC). Acest modul constă dintr-un driver de motor L298 IC și un regulator 78M05 5V. Modulul L298N poate controla până la 4 motoare DC sau 2 motoare DC cu controlul direcției și al vitezei.

3. Implementarea

Implementarea acestui robot de urmărire a obiectelor colorate este format din componenta mecanică, electrică și de firmware.

A. Structura mecanică

Șasiu

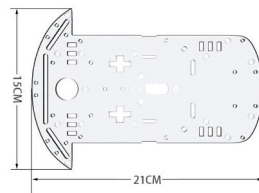


Fig.2. Șasiu

Roți



Fig.3. Roți

Motoare cu reductor - 3-6V

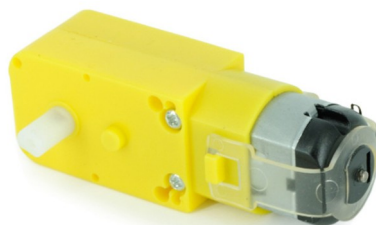
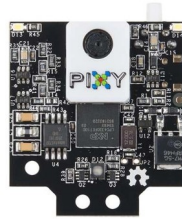


Fig.4. Motor cu reductor DC

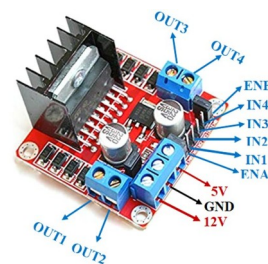
B. Partea electronică și hardware

Camera Pixy2



Processor	NXP LPC4330, 204 MHz, Dual-core
Image sensor	Aptina MT9M114, 1296×976 resolution with integrated image flow processor
Lens field-of-view	60 degrees horizontal, 40 degrees vertical
Power consumption	140 mA typical
Power input	USB input (5V) or unregulated input (6V to 10V)
RAM	264K bytes
Flash	2M bytes
Available data outputs	UART serial, SPI, I2C, USB, digital, analog

Driver motoare - L298N



BLOCK DIAGRAM

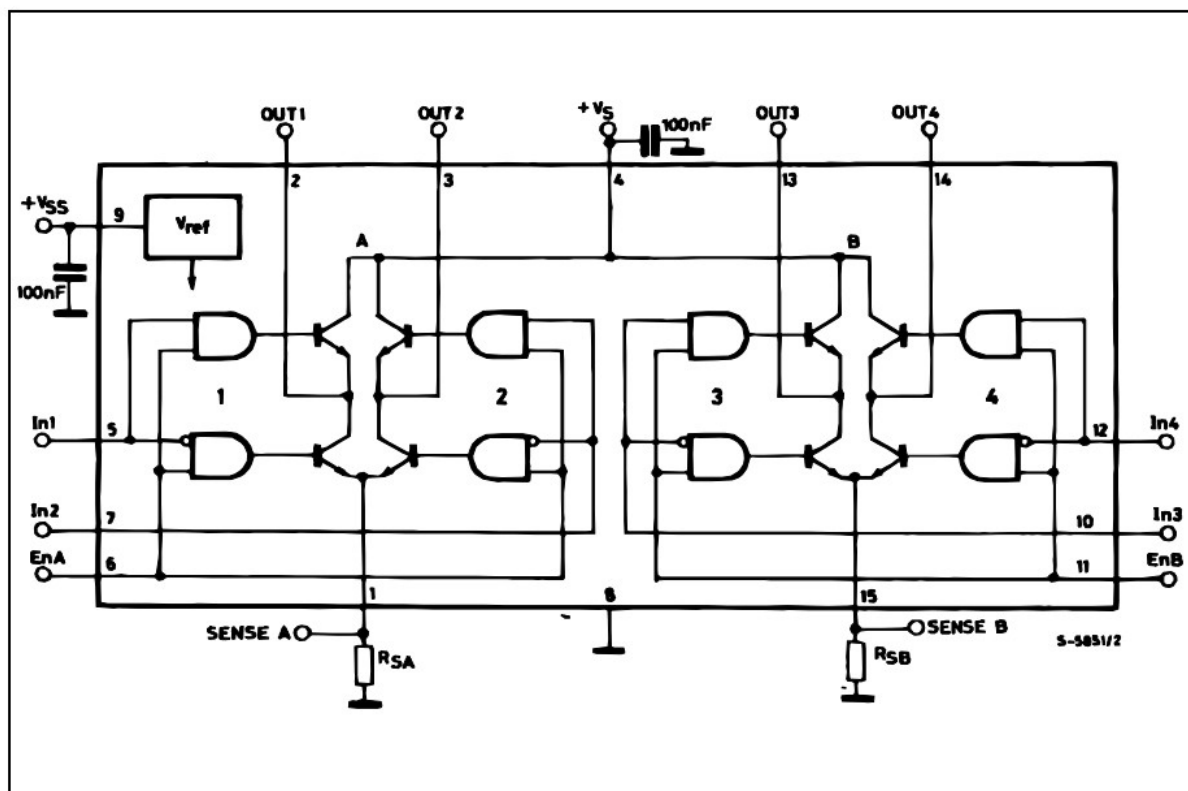
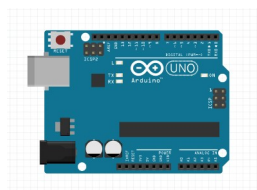


Fig.5. Diagrama bloc L298

Specificații tehnice:

- Motor voltage: 5 V - 35 V
- Logic circuit voltage: 5 V
- Motor current: 2 A (MAX)
- Logic current: 36 mA
- Maximum frequency pwm: 40 kHz



Arduino Uno R3

Microcontroller	ATmega328P	
Pins	Built-in LED Pin	13
	Digital I/O Pins	14
	Analog input pins	6
	PWM pins	6
Communication	UART, I2C, SPI	
Power	I/O Voltage	5V
	Input voltage (nominal)	7-12V
	DC Current per I/O Pin	20 mA
Clock speed	Main Processor	ATmega328P 16 MHz
	USB-Serial Processor	ATmega16U2 16 MHz
Memory	ATmega328P	2KB SRAM, 32KB FLASH, 1KB EEPROM

Sursă de alimentare

2 seturi a câte 4 baterii AA (1.5 V) conectate în serie = 12 V

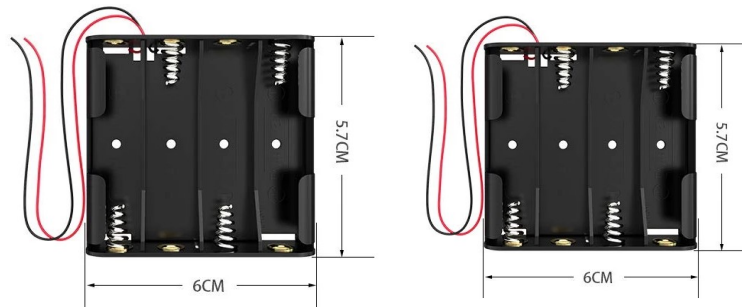


Fig.6. Suport baterii

Design hardware

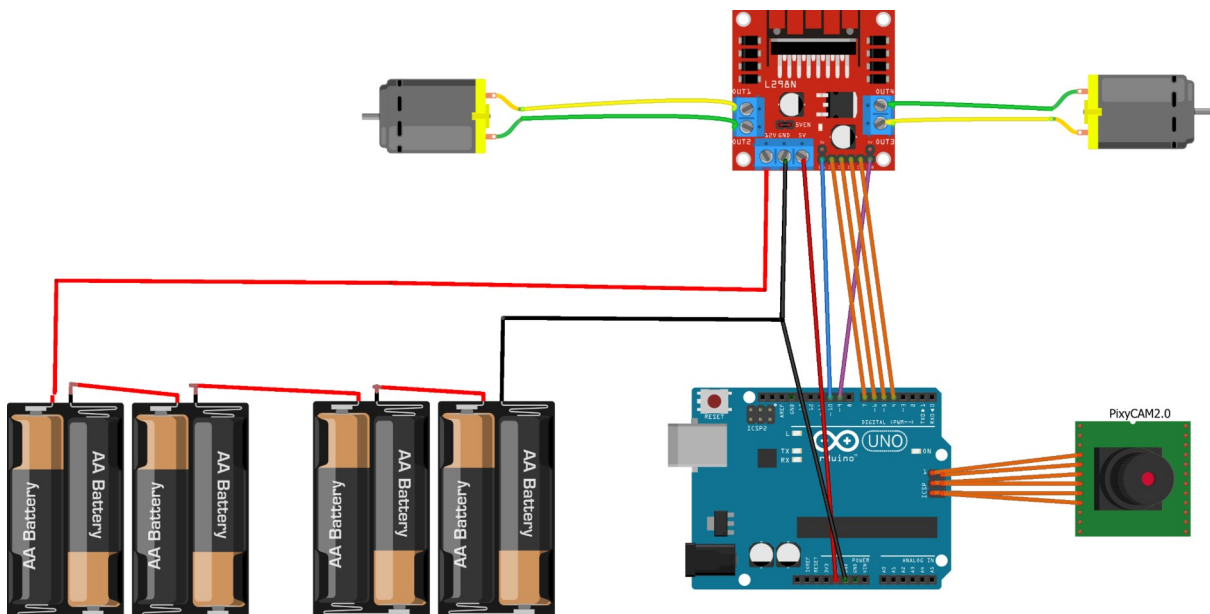


Fig.7.Design Hardware

C. Design Firmware

Procesul se începe prin inițializarea camerei și plăcii Arduino.

Se face verificarea unei funcționări stabile a camerei și a plăcii la parametrii actuali a sistemului și a conexiunii celor două componente prin portul ICSP.

După aceea se face detectarea obiectelor, primirea datelor și selectarea celor necesare: lățimea obiectului și poziția x. Valoarea lățimii reprezintă distanța dintre obiect și cameră, iar valoarea x reprezintă poziția obiectului față de robot pe axa Ox.

Cu cât lățimea obiectului este mai mică, distanța până la obiect este mai mare și invers.

Robotul tinde să se afle în stare de repaus, adică obiectul trebuie să fie centrat pe axa Ox și să se afle la o distanță de referință predefinită de programator.

- Obiectul este centrat, dacă se află în intervalul de valori 150 și 180 pe Ox.
- Distanța de referință setată este aproximativ de 15cm între obiectul urmărit și cameră, care corespunde intervalului de valori 70-110 reprezentată prin lățimea dreptunghiului ce selectează obiectul urmărit.






L \ X	0 ... 149	150 ... 180	181...316
0 ⋮ 69			
70			

Fig.8. Reprezentarea abstractă a vectorului de mișcare

Dacă valoarea este mai mică ca 150, virează la stânga, iar dacă e mai mare ca 180, virează la dreapta.

Virarea robotului se face prin setarea unei viteze mai lente pentru una dintre roți. Cu cât obiectul e mai la dreapta de robot, cu atât acesta va vira mai intens, analog pentru virarea la stânga.

Dacă lățimea obiectului este între 70 și 110, atunci robotul va trece în stare de repaus.

Dacă lățimea acestuia este mai mare de 110, acesta se va deplasa în spate, iar dacă mai mic 70, se va deplasa înainte.

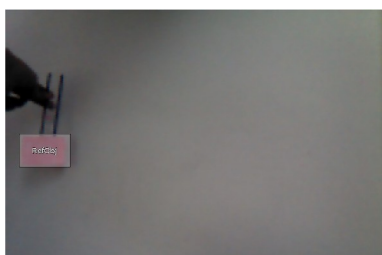


Fig.9. Experiment detecția obiectului la stânga

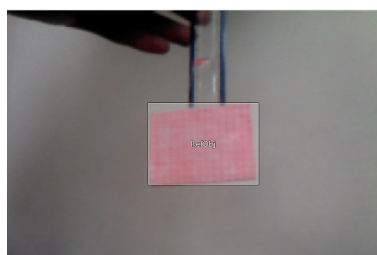


Fig.10. Experiment detecția obiectului pe centru

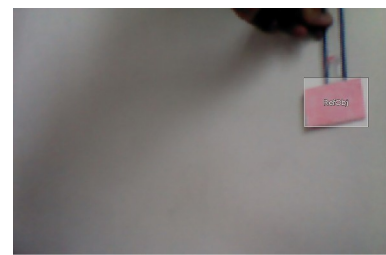


Fig.11. Experiment detecția obiectului la dreapta.

Algoritmi de procesare a datelor

PID și Interpolare

PID controller.

Proportional–Integral–Derivative Controller - calculează în continuu valoarea de eroare ca diferență dintre valoarea dorită și valoarea procesată, apoi aplică o corecție bazată pe componenta Proportională, Integratoare și Derivativă.

```
float KP = 1.25;    // coeficientul pentru comp. proportionala
float KI = 0.0001; // coeficientul pentru comp. integratoare
float KD = 0.85;    // coeficientul pentru comp. derivativă
```

```
error = normal_x - obj_x; // calculăm eroarea
P = error;
I = I + error;
D = error - lastError;
lastError = error;

float motorspeed = P * KP + I * KI + D * KD;
```

Fig.12. Implementarea PID-ului în cod

Se calculează eroarea.

Componenta proportionala este însuși eroarea.

Componenta integratoare este eroarea cumulată.

Componenta derivativă este diferența dintre ultima eroare și cea curentă.

Se calculează viteza.

Interpolarea polinomială.

În analiză numerică, interpolarea polinomială este o tehnică de interpolare a unui set de date sau a unei funcții printr-un polinom.

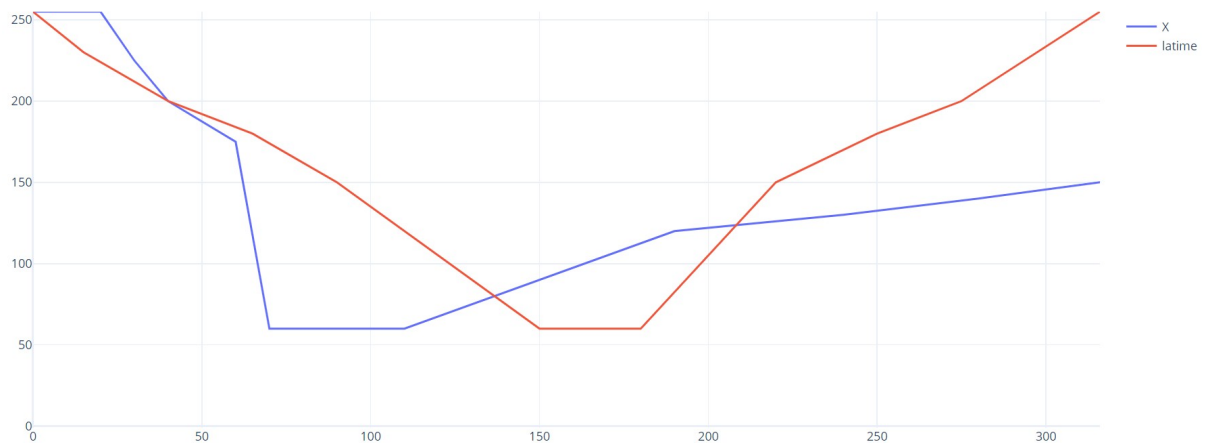


Fig.13. Graficul funcțiilor utilizate pentru formarea semnului final.

Formula de interpolare liniară pentru situația unui tabel de interpolare unidimensional este:

$$R = R_{i-1} + \frac{V - V_{i-1}}{V_i - V_{i-1}} (R_i - R_{i-1})$$

4. Bibliografie

1. https://media.digikey.com/pdf/Data%20Sheets/Seed%20Technology/102991074_Web.pdf
2. https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf
3. <https://docs.arduino.cc/hardware/uno-rev3>
4. <https://www.teachmemicro.com/arduino-pid-control-tutorial/>
5. <https://www.youtube.com/watch?v=4Y7zG48uHRo&t=96s>
6. <https://alexgyver.ru/lessons/filters/>
7. <https://alexgyver.ru/gyverpid/>
8. <https://www.electronicwings.com/users/sanketmallawat91/projects/215/frequency-changing-of-pwm-pins-of-arduino-uno>
9. <https://forum.arduino.cc/t/linear-interpolation-using-2d-lookup-table/406595>