

OLIMPIADA NAȚIONALĂ DE CREATIVITATE ȘTIINȚIFICĂ

SECȚIUNE - ȘTIINTE APLICATE
CATEGORIA JUNIORI

DENUMIRE PROIECT

**Sistem robotic pentru intervenție în cazul situațiilor de
urgenta**

Numele autorilor:

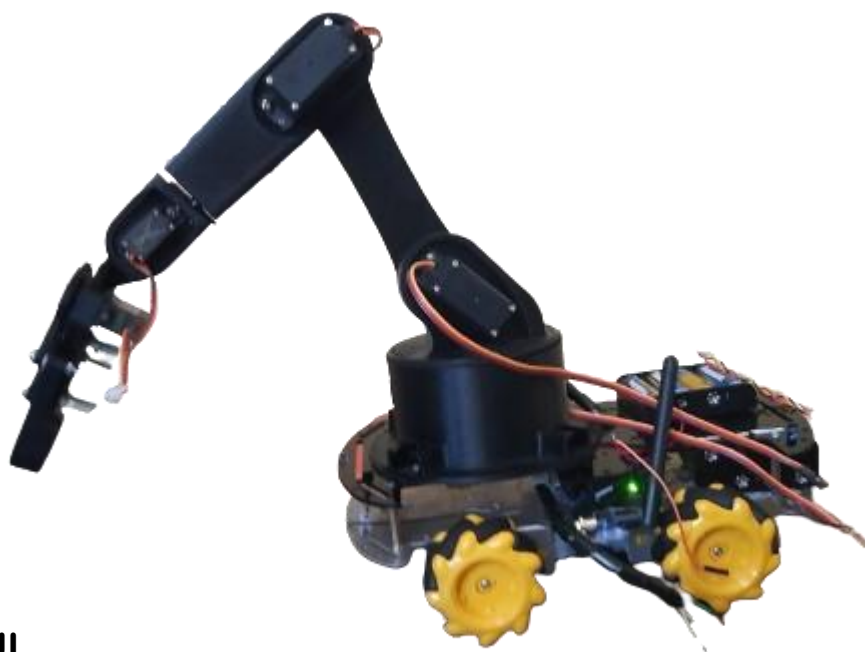
**Lider Chitic Rares-Adrian
Stanciu Matei**

Nume coordonatori : profesor Chirițescu Ana - Carmen

Centrul Județean de Excelență Vaslui

Liceul Teoretic Mihail Kogălniceanu Vaslui

Școala Gimnazială “Constantin Parfene” Vaslui.



SCOPUL

Statisticile au arătat faptul că în ultimii ani numărul victimelor datorate producerii situațiilor de urgență, precum cutremure, explozii, inundații, alunecări de teren sau

incendii, este în creștere. Astfel, este necesară utilizarea sistemelor robotice care pot asista echipele de intervenție în eforturile de salvare a vieților. Pe lângă operațiunile de căutare și salvare, roboții pot juca și un rol esențial în alte aspecte ale gestionării situațiilor de urgență, cum ar fi asigurarea logisticii și transmiterea informațiilor de la locul producerii evenimentului în timp real sporind eficiența operațiunilor de intervenție

OBIECTIVE

Roboții pot fi considerați asistenți în cazul intervențiilor în situații de urgență, deoarece pot monitoriza de la distanță sau să intervină în zone inaccesibile sau care pun viața în pericol. Ei pot fi utilizați pentru transportul de provizii, medicamente, oxigen și echipament sau cu ajutorul senzorilor și camerelor pot localiza persoane și pot transmite informații despre gravitatea situației către echipele de salvare, din apropierea dezastrului, fără a pune în pericol viața salvatorilor profesioniști. Aceste informații pot fi ulterior utilizate pentru conceperea planului de intervenție și gestionarea situației într-un mod mai eficient.

MOD DE OPERARE

Sistemul robotic pentru intervenția în cazul situațiilor de urgență, poate fi operat manual prin control de la distanță folosind o aplicație prin care poți genera comenzi prin vederea directă a spațiului datorită camerei și a locației exacte a robotului.

Pe viitor am putea instala și alte componente, cum ar fi:

- Modul GPS(NEO-6M)
- Senzor PIR(senzor prezenta umana bazat pe temperatura)
- Suspensii
- Card SD 4 Gb pentru stocarea imaginilor captate de camera video

PROBLEMA IDENTIFICATA SPRE REZOLVARE

Robotul rezolvă problema eficienței sistemelor de intervenție. Acesta ajută la minimizarea timpului de expertiză oferind date în timp real către echipajele specializate, astfel ajută și la protecția persoanelor implicate nefiind supuse la riscul unor situații periculoase. Acesta poate ajuta și la căutări pe raze mici fiind echipat cu o cameră echipată cu lumină. Rotile mecanice și tracțiunea integrală îi oferă mișcări rapide și eficiente pentru a minimiza maxim timpul de intervenție.

PROBLEME ÎNTALNITE ÎN REALIZAREA ROBOTULUI:

- Din cauza faptului că funcțiile `bt.read()` și `Serial.read()` trimit date de până la 1 byte a fost complicat de găsit o logică pentru controlul servomotoarelor. Am rezolvat această problemă implementând o logică formată din trimiterea de numere sub 1 byte ce incrementează sau decrementează valorile unghiurilor servo Motoarelor.
- Robotul avea un timp de reacție relativ mare la comenzile trimise prin Bluetooth. Acest fapt a fost datorat folosirea multiplilor instrucțiuni de decizie în program reducând numărul maxim de verificări de la 23 la o singură verificare. Am reușit acest lucru folosind în loc de o instrucțiune `if` o instrucțiune `switch` cu multiple cazuri.

- Am diminuat erorile si deconectările aleatorii folosind biblioteca SoftwareSerial.h ce generează o serie de pini tx rx (in cazul nostru pinii 9 si 10).Acesta a redus erorile deoarece nu a mai folosit pini prestabiliți 0 si 1 ce sunt prestabiliți ca pini seriali.

ETAPE PARCURSE

- REALIZAREA CIRCUITULUI ELECTRIC
- ASAMBLAREA COMPONENTELOR PE SASIU
- ASAMBLAREA BRATULUI
- PRINDEREA BRATULUI
- SCRIEREA CODULUI
- CREEREA DESIGNULUI APLICATIEI
- SCRIEREA CODULUI APLICATIEI
- TESTAREA ROBOTULUI
- OPTIMIZAREA CODULUI SI A APLICATIEI

• METODE FOLOSITE

Poziția ideală de început a brațului

Conform formulei: Momentul forței este egal cu forța(ce acționează pe ax) înmulțita cu axul forței ,poziția ideală pentru a se consuma cat de putina energie posibila este aceea in care toate axele sunt coliniare pe axul de rotație al axei a 2-a .Astfel brațul forței ar fi tehnic 0 ce ar determina ca momentul forței sa fie in urmare 0.In cazul in care axa 3 ar fi orientata in fata robotului centrul de greutate al robotului s ar schimba iar acest fapt poate impacta negativ performanta robotului.

Transmiterea datelor prin Bluetooth

Transmiterea datelor prin Bluetooth este procesul de trimitere a informațiilor între dispozitive utilizând tehnologia Bluetooth, care folosește unde radio pentru a crea conexiuni fără fir între aceste dispozitive. Acest proces implică cel puțin două dispozitive Bluetooth care pot fi, de exemplu, un smartphone, o tabletă, un calculator sau o placă Arduino. Transmiterea datelor prin Bluetooth poate implica diverse tipuri de date, cum ar fi text, fișiere, semnale de control, date de senzori și multe altele.

Principiul de funcționare a roților mecanum

Principiul de bază al funcționării roților mecanum este utilizarea unei configurații speciale a roților. Rolurile sunt montate într-o anumită configurație pe suprafața roții, în general la un unghi de 45 de grade față de direcția de deplasare a vehiculului. Aceasta înseamnă că atunci când roata mecanum se rotește, rolele se rotesc la un unghi diferit, permițând deplasarea vehiculului într-o direcție laterală.

Atunci când roțile mecanum sunt montate pe un vehicul în configurație adecvată și sunt acționate corespunzător, ele permit acestuia să se deplaseze înainte, înapoi, lateral și să efectueze rotații în jurul propriului ax. Această versatilitate face ca roțile mecanum să fie ideale pentru aplicații în care manevrabilitatea este crucială, precum robotică mobilă, manipulare de precizie sau vehicule autonome.

- **Date experimentale si detalii esențiale ale experimentelor**

PIESE NECESARE

În acest proiect am folosit următoarele piese:

- Arduino uno
- PCA9685
- 3 buc servomotor MG90S
- 3 buc servomotor MG996S
- Shield control motoare(4 canale) L293D
- ESP-32-CAM
- Antena 2G
- Router wifi
- Sașiu metalic OKY5030(aluminiu)
- Sașiu de plastic
- 4 buc motor DC cu reductor
- Roti omnidirecționale/meccanum
- Roti normale
- Schelet braț robotic
- Fire dupont
- Telefon/Tableta
- cutie baterii 6V
- Cutie de baterii 12V
- Fir micro-usb
- 4 buc Extensii de fire



COD

Acesta este codul pe care l-am utilizat pentru robot:

```
#include <AFMotor.h>
#include <Adafruit_PWMServoDriver.h>
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
SoftwareSerial bt(9, 10);
```

```
AF_DCMotor motor1(1); // Motorul stânga
față
```

```

AF_DCMotor motor2(2); // Motorul stanga
spate
AF_DCMotor motor3(3); // Motorul
dreapta fata
AF_DCMotor motor4(4); // Motorul
dreapta spat

```

```

Adafruit_PWMServoDriver pwm;
void Servo_pos(int servoNum, int unghi) {
  if (unghi < 0 || unghi > 180) return;
  if (servoNum==5 and unghi < 0 || unghi >
90) return;
  int pulse = map(unghi, 0, 180, 150, 600);
  pwm.setPWM(servoNum, 0, pulse);
  Serial.print(servoNum);
  Serial.print(" ");
  Serial.println(unghi);
}

```

```

void stop() {
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}

```

```

void fata() {
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}

```

```

void spat() {
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
}

```

```

void stanga() {
  motor1.run(BACKWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(BACKWARD);
}

```

```

void dreapta() {
  motor1.run(FORWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(FORWARD);
}

```

```

void fata_stanga() {
  motor1.run(RELEASE);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(RELEASE);
}

```

```

void fata_dreapta() {
  motor1.run(FORWARD);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(FORWARD);
}

```

```

void spat_stanga() {
  motor1.run(BACKWARD);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(BACKWARD);
}

```

```

void spat_dreapta() {
  motor1.run(RELEASE);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
}

```

```

  motor4.run(RELEASE);
}
void rotire_stanga() {
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}
void rotire_dreapta() {
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
}
int data;
int pos1 = 90, pos2 = 90, pos3 = 80, pos4 =
90, pos5 = 90, pos6 = 90;
void setup() {
  Serial.begin(9600);
  bt.begin(9600);
}

```

```

  pwm.begin();
  pwm.setPwmFreq(60);
  pinMode(9, INPUT);
  pinMode(10, OUTPUT);
  motor1.setSpeed(255); // Seteaza viteza
motorului
  motor2.setSpeed(255);
  motor3.setSpeed(255);
  motor4.setSpeed(255);
}

```

```

  Servo_pos(0, pos1);
  Servo_pos(1, pos2);
  Servo_pos(2, pos3);
  Servo_pos(3, pos4);
  Servo_pos(4, pos5);
  Servo_pos(5, pos6);
}

```

```

void loop() {
  if (bt.available()) {
    data = bt.read();
    switch (data) {
      case 0:
        stop();
        break;
      case 1:
        fata();
        break;
      case 2:
        spat();
        break;
      case 3:
        stanga();
        break;
      case 4:
        dreapta();
        break;
      case 5:
        fata_stanga();
        break;
      case 6:
        fata_dreapta();
        break;
      case 7:
        spat_stanga();
        break;
      case 8:
        spat_dreapta();
        break;
    }
  }
}

```

```

case 9:
  rotire_stanga();
  break;
case 10:
  rotire_dreapta();
  break;
case 11:
  pos1--;
  Servo_pos(1, pos1);
  delay(3);
  break;
case 12:
  pos1++;
  Servo_pos(1, pos1);
  delay(3);
  break;
case 13:
  pos2--;
  Servo_pos(1, pos2);
  delay(3);
  break;
case 14:
  pos2++;
  Servo_pos(2, pos2);
  delay(3);
  break;
case 15:
  pos3--;
  Servo_pos(3, pos3);
  delay(3);
  break;
case 16:
  pos3++;
  Servo_pos(3, pos3);
  delay(3);
  break;
case 17:
  pos4--;
  Servo_pos(4, pos4);
  delay(3);
  break;
case 18:
  pos4++;
  Servo_pos(4, pos4);
  delay(3);
  break;
case 19:
  pos5--;
  Servo_pos(5, pos5);
  delay(3);
  break;
case 20:
  pos5++;
  Servo_pos(5, pos5);
  delay(3);
  break;
case 21:
  pos6--;
  Servo_pos(6, pos6);
  delay(3);
  break;
case 22:
  pos6++;
  Servo_pos(6, pos6);
  delay(3);
  break;
}
}

```

DESPRE PIESE PLACA ARDUINO UNO



Arduino Uno este o placă de dezvoltare open-source, bazată pe un microcontroler ATmega328P, dezvoltată de Arduino LLC. Aceasta este una dintre cele mai populare plăci de dezvoltare pentru prototipare rapidă, programare și controlul sistemelor electronice și este utilizată într-o varietate de aplicații, de la proiecte de hobby la aplicații comerciale și industriale.

Placa Arduino Uno poate fi programată cu ajutorul software-ului Arduino IDE, care este ușor de utilizat și permite dezvoltatorilor să scrie și să încarce cod pe placa Arduino. De asemenea, placa dispune de o serie de librării și exemple de cod, care facilitează dezvoltarea de aplicații pentru diferite proiecte.

Shield L293D

L293D este un circuit integrat (IC) utilizat pentru controlul motoarelor și acționarea altor dispozitive electrice. Este folosit adesea pentru a controla direcția și viteza motoarelor în aplicații cum ar fi roboții, mașinuțe cu telecomandă, sisteme de control automat și multe altele. Acest IC este un circuit foarte versatil și este adesea folosit în proiecte electronice.



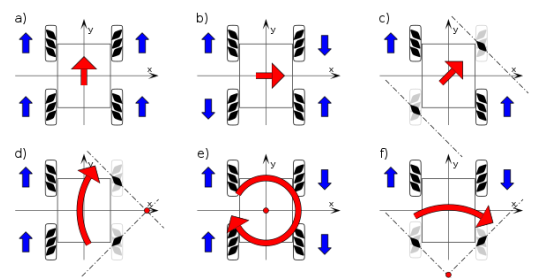
Caracteristicile principale ale L293D includ:

Controlul motorului: L293D permite controlul a doi motoare DC (motoare cu curent continuu) sau a unui motor pas cu pas. Acesta poate controla direcția de rotație a motoarelor și poate modula viteza lor.

Supraîncălzire și protecție împotriva suprasarcinii: L293D are o protecție împotriva supraîncălzirii și a suprasarcinii pentru a preveni deteriorarea IC-ului în timpul utilizării prelungite la sarcini mari.

Roti omnidirecționale/mecanum

Rotile omnidirecționale, cunoscute și sub numele de roți Mecanum, sunt un tip special de roți folosite în roboți și alte vehicule pentru a permite mișcarea în orice direcție, fără a necesita o rotație a întregului vehicul. Aceste roți sunt dotate cu role speciale, care sunt montate diagonal pe suprafața lor. Aceste role sunt înglobate într-un unghi specific, ceea ce permite roților să se rotească în jurul axei lor individuale.



Principiul de funcționare al roților omnidirecționale se bazează pe utilizarea forțelor de fricțiune. Atunci când roțile se rotesc la viteze diferite sau în direcții diferite, forța de fricțiune rezultată între role și suprafața de rulare determină mișcarea într-o direcție specifică. Astfel, prin controlul independent al fiecărei roți, vehiculul poate efectua mișcări complexe, cum ar fi derapajul lateral, rotația la loc sau mișcări diagonale.

SERVO MOTOR TOWER PRO MG90S

Servo motorul Tower Pro MG90S este un motor DC cu curent continuu și un controler încorporat, care îi permite să se rotească la un unghi specific și să mențină poziția respectivă.



Motorul Tower Pro MG90S are un pin de semnal, un pin de alimentare și un pin de masă pentru conectarea la placa Arduino sau alt controler. Controlul servo motorului se realizează prin trimiterea unui semnal PWM (pulse-width modulation) către pinul de semnal, care determină poziția unghiulară a motorului.

SERVO MOTOR TOWER PRO MG996R

Servomotorul MG996R este un tip popular de servomotor utilizat în aplicații de robotică, controlul mișcării și alte proiecte electronice. Este un servomotor de mare putere, fiabil și accesibil ca preț. Iată câteva caracteristici principale:

Precizie: Este cunoscut pentru precizia sa în poziționare și mișcare, făcându-l potrivit pentru proiecte care necesită control fin al poziției.

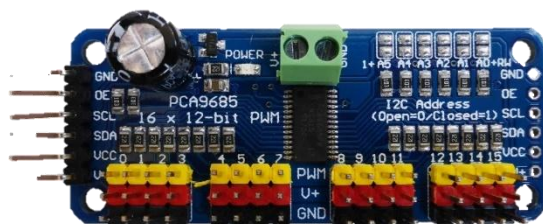
Tensiune de operare: Funcționează la o tensiune de alimentare de obicei între 4,8V și 6,6V, ceea ce îl face compatibil cu majoritatea surselor de alimentare disponibile.



PCA 9685

PCA9685 este un circuit integrat specific care este adesea folosit ca un driver pentru dispozitive periferice, cum ar fi LED-uri sau servo-motoare în aplicații de control.

Driverul PCA9685 permite controlul individual al fiecărui canal PWM, permițând astfel reglarea intensității luminii LED-urilor sau poziției servo-motoarelor. Acesta suportă diferite frecvențe PWM și are funcționalități suplimentare, cum ar fi posibilitatea de a sincroniza mai mulți drivere PCA9685 pentru a extinde numărul de canale disponibile.



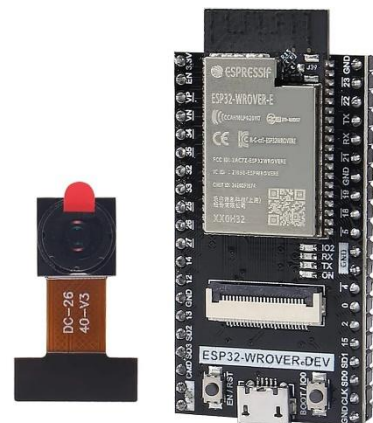
Modul ESP32 CAM cu camera OV2640

ESP32-CAM Este un modul de camera de dimensiuni reduse, cu consum redus de energie, bazat pe ESP32. Este un modul complet cu un microcontroller integrat, care îl poate face să funcționeze independent. Pe lângă conectivitatea WiFi + Bluetooth, acest modul are și o cameră video integrată OV2640 și un slot microSD pentru stocare.

Motor DC cu reductor

Acesta este un mini motor DC cu reducere, ideal pentru **fabricarea de roboți**, deoarece are o greutate redusă, cuplu mare și turație scăzută. Utilizat pe scară largă în robotică, și control al mișcării liniare, etc.

Concluzii



În concluzie, considerăm că în gestionarea situațiilor de urgență, utilizarea sistemelor robotice combinat cu experiența echipelor de primă intervenție, poate îmbunătăți semnificativ capacitatea de a răspunde la dezastrele naturale și de a salva vieți.

Rolul membrilor echipei de proiect

CHITIC RARES	STANCIU MATEI
CONCEPTUL SISTEMULUI	AFIS
COD	REFERAT
CIRCUIT ELECTRIC	IMPRIMARE 3D
COD APLICATIE	
ASAMBLARE	
CUMPARARE COMPONENTE	
DESIGN APLICATIE	

Credite si mențiuni

Proiect realizat de elevii Liceului Teoretic Mihail Kogălniceanu Vaslui, Scolii Gimnaziale Constantin Parfene Vaslui si Centrului Județean de Excelenta Vaslui.

Chitic Rares-Adrian, clasa: a-VII-a - lider

Stanciu Matei, clasa: a-VII-a

Coordonator: Prof.Chiritescu Ana-Carmen