

CĂLĂTORIE SPRE ROMÂNIA!

prof. Călugăru Dumitru
Disciplina: TIC

Titlu	Călătorie spre România!	Timp: 2 ore
Subiect:	extracurricular	

Obiecte:	Competență generală ¹ : 1. Exersarea gândirii computaționale în diferite contexte de programare a roboților reali Competență specifică ² : 1.1 Testarea algoritmilor pentru operarea dispozitivelor robotizate pentru mișcarea spațială Competență specifică ³ 1.2 Utilizarea algoritmilor în diferite situații pentru codificarea roboților Scopul activității : Codificarea unui robot care, cu ajutorul senzorilor de culoare, parcurge un traseu simplu, prestabilit.	
Elemente cheie de gândire computațională:	Descompunere; Recunoașterea modelelor; Abstractizarea; Proiectarea algoritmului;	
Grupa de vârstă	12-14 ani	
Situații de învățare:	Sala de clasă	Tip activitate:
Resurse:	Truse de robot (microcontroller, senzori, motoare, servomotoare, relee, porturi, actuatoare etc.), carte tehnica, specificatii kit, schema functionala, laptop-uri, traseu, steaguri tarilor partenere.	

¹Conform Curriculumului Național de Educație Artă, clasa a VI-a, aprobat prin Ordinul Ministrului Educației Naționale nr. 3393 / 28.02.2017.

²Idem

³Idem

Dezvoltare

Definirea problemei:

- Profesorul le spune elevilor că munca constă în scrierea și testarea algoritmilor folosiți pentru a controla mișcările unui robot în diferite direcții și pentru a înțelege valorile citite de senzori. Elevii vor identifica algoritmi folosiți pentru operarea mecanismelor unui robot (ex. urmărirea liniei - calea prestabilită - pentru a finaliza sarcina „Călătorie în România”).
- Elevii vor lucra în echipă. Fiecare echipă va codifica robotul pentru a călători din țara de origine (Turcia, Spania, Italia) în România.

eu . Descompunere:

1. Analizați, explicați și înțelegeți algoritmi utilizați în proiectarea unui robot, ca pas distinct înainte de proiectarea acestuia.

- a. Sarcinile presupun alegerea destinației, stabilirea rutei optime, urmărirea traseului/traseului, revenirea la traseu dacă sunt oscilații de mișcare, oprirea mișcării când a fost atinsă destinația.
- b. Schimbarea regulilor de călătorie dacă este necesar din cauza condițiilor cu care se confruntă.

2. Scrierea algoritmului de programare a robotului

- a. Fiecare echipă va decide, în funcție de componentele puse la dispoziție, geometria unui robot, funcționalitatea componentelor, circuitele electronice să fie alcătuite din componente pasive și active.
- b. Legarea blocurilor funcționale cu un algoritm care corelează utilizarea lor cu calea de a ajunge la destinație va avea ca rezultat un program reflectat de codul sursă. Codul sursă va fi rulat de platforma Arduino, inclusiv un procesor ESP 32.

3. Scrierea programului, a codului sursă

Având în vedere numărul de senzori, numărul de motoare de propulsie și controlul acestora prin intermediul informațiilor senzorilor, rezultă următorul cod sursă:

```
// Codul robotului Arduino Line Follower
```

```
#define enA 5//Enable1 L293 Pin enA
#define in1 6 //Motor1 L293 Pin in1
#define in2 7 //Motor1 L293 Pin in1
#define in3 9 //Motor2 L293 Pin in1
#define in4 10 //Motor2 L293 Pin in1
#define enB 8 //Enable2 L293 Pin enB
#define Senzor R_S 4//ir Dreapta
#define L_S 2 //senzor ir Stânga
```

CĂLĂTORIE SPRE ROMÂNIA!

prof. Călugăru Dumitru
Disciplina: TIC

```
void setup(){
pinMode(R_S, INPUT);

pinMode(L_S, INPUT);
pinMode(enA, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
pinMode(enB, OUTPUT);
digitalWrite(enA, HIGH);
digitalWrite(enB, HIGH);
întârziere(1000);}

void loop(){

if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0)){forward();}
if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 0)){turnRight();}
if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)){turnLeft();}
if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 1)){Stop();}
}

void înainte(){

digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);}
void turnDreapta(){
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
```

```
digitalWrite(in4, HIGH); }  
  
void turnLeft(){  
  
digitalWrite(in1, HIGH);  
digitalWrite(in2, LOW);  
digitalWrite(in3, HIGH);  
digitalWrite(in4, LOW); }  
  
void Stop(){  
  
digitalWrite(in1, LOW);  
digitalWrite(in2, LOW);  
digitalWrite(in3, LOW);  
  
digitalWrite(in4, LOW);  
}  
}
```

4. Parcurgerea traseului prestabilit

Pentru a urma traseul, conectați robotul prin cablu USB la computer, conectați COM corespunzător, conectați procesorul, verificați codul și apoi încărcați-l pe platforma Arduino. După ce ați încărcat codul, plasați robotul pe cale și urmăriți progresul acestuia.

II . Recunoașterea modelelor: traseul, roboți

Fiecărui grup i se oferă kiturile, se construiesc roboții, se trasează rutele dintre punctele de plecare și destinația stabilită.

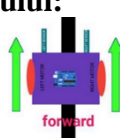
III. Abstracție:

Profesorul subliniază semnificația fiecărei etape.

Dacă elevii scriu algoritmul corect, atunci robotul va urma traseul planificat.

IV . Proiectarea algoritmului:

1. Mergând înainte



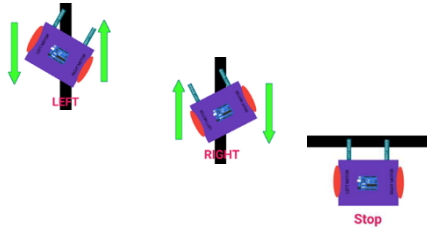
CĂLĂTORIE SPRE ROMÂNIA!

prof. Călugăru Dumitru
Disciplina: TIC

2. Viraj la stânga

3. Obligatoriu Dreapta

4. Stop



Evaluare:

Analizarea performanței robotului pentru diferite intrări

Rezultate așteptate:

Proiectarea corectă a robotului, programarea acestuia, rutarea corectă.