



Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

PORTOFOLIU DE PRACTICĂ

MOBILITATE CIPRU

10.05.2026 – 23.05.2026

IACOB VICTOR

LEUȘTEAN MARIA

PASCU ANDREI

DISCLAIMER: „Finanțat de Uniunea Europeană. Punctele de vedere și opiniile exprimate aparțin, însă, exclusiv autorului (autorilor) și nu reflectă neapărat punctele de vedere și opiniile Uniunii Europene sau ale Agenției Naționale pentru Programe Comunitare în Domeniul Educației și Formării Profesionale (ANPCDEFP). Nici Uniunea Europeană și nici ANPCDEFP nu pot fi ținute răspunzătoare pentru acestea.”



Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

Cuprins

PROIECT: Traffic Barrier	2
PROIECT: Conveyor Stepper	8
PROIECT FINAL: Car Presentation Showroom	10



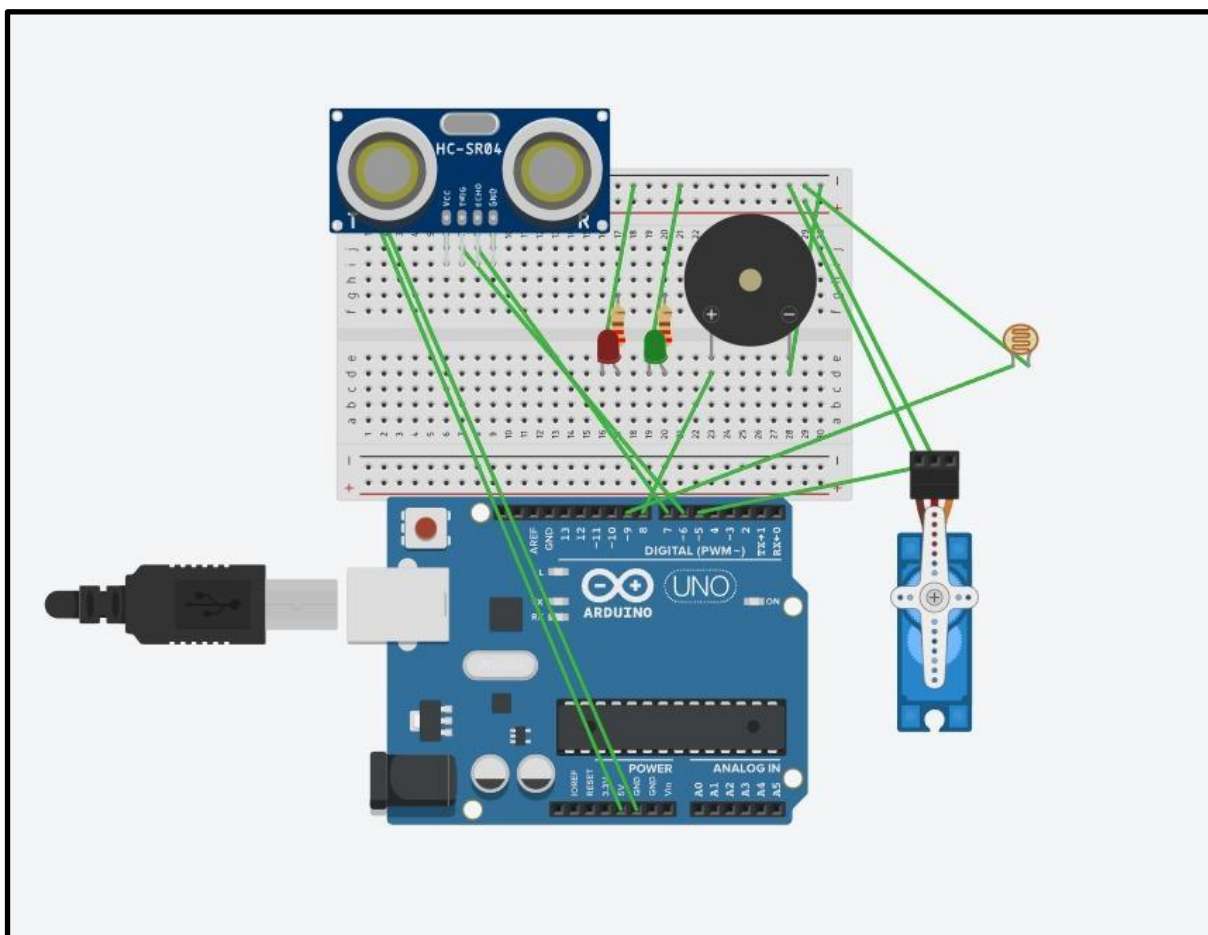
Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

PROIECT: Traffic Barrier

Schematic:



This project shows how a traffic barrier works.

Components:

Breadboard;

Arduino R3;

Photoresistor sensor;

Ultrasonic sensor;

Servomotor;

Buzzer;



Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

Leds;

Wires;

Resistors(220Ω);

Utilities:

-it can be used in private parking lots;

-in frontiers;

Advantages:

-safe;

-easy to enter in private lots;

CODE :

```
#include <Servo.h>

Servo motor;

int trigPin=11;

int echoPin=12;

int duration;

int cm;

int verde=5;

int rosu=4;

int sensor=A0;

int val;

int buzzer=7;
```



Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

```
void setup() {  
  
    pinMode(trigPin, OUTPUT);  
  
    pinMode(echoPin, INPUT);  
  
    pinMode(rosu, OUTPUT);  
  
    pinMode(verde, OUTPUT);  
  
    pinMode(sensor, INPUT);  
  
    pinMode(buzzer, OUTPUT);  
  
    Serial.begin(9600);  
  
    motor.attach(8);  
  
}  
  
void loop() {  
  
    val=analogRead(sensor);  
  
    digitalWrite(trigPin, LOW);  
  
    delayMicroseconds(5);  
  
    digitalWrite(trigPin, HIGH);  
  
    delayMicroseconds(10);  
  
    digitalWrite(trigPin, LOW);  
  
  
    pinMode(echoPin, INPUT);  
  
    duration=pulseIn(echoPin, HIGH);  
  
  
    cm=(duration/2)/29.1;
```



Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

```
if(cm<0 || cm>150) {Serial.print("Prea aproape / Prea departe");  
Serial.println();}  
  
else{  
  
Serial.print("Distance: ");  
  
Serial.print(cm);  
  
Serial.print(" cm");  
  
Serial.print("                ");  
  
Serial.print("Brightness: ");  
  
Serial.print(val);  
  
Serial.println();  
}  
  
if(cm>20 || val>350){  
  
digitalWrite(rosu,HIGH);  
  
digitalWrite(verde,LOW);  
  
noTone(buzzer);  
  
motor.write(45);  
  
delay(1000);  
}  
  
else if(cm<20 && val<350){  
  
digitalWrite(rosu,LOW);  
  
digitalWrite(verde,HIGH);  
  
tone(buzzer,1500);  
  
motor.write(130);  
  
delay(3500);
```

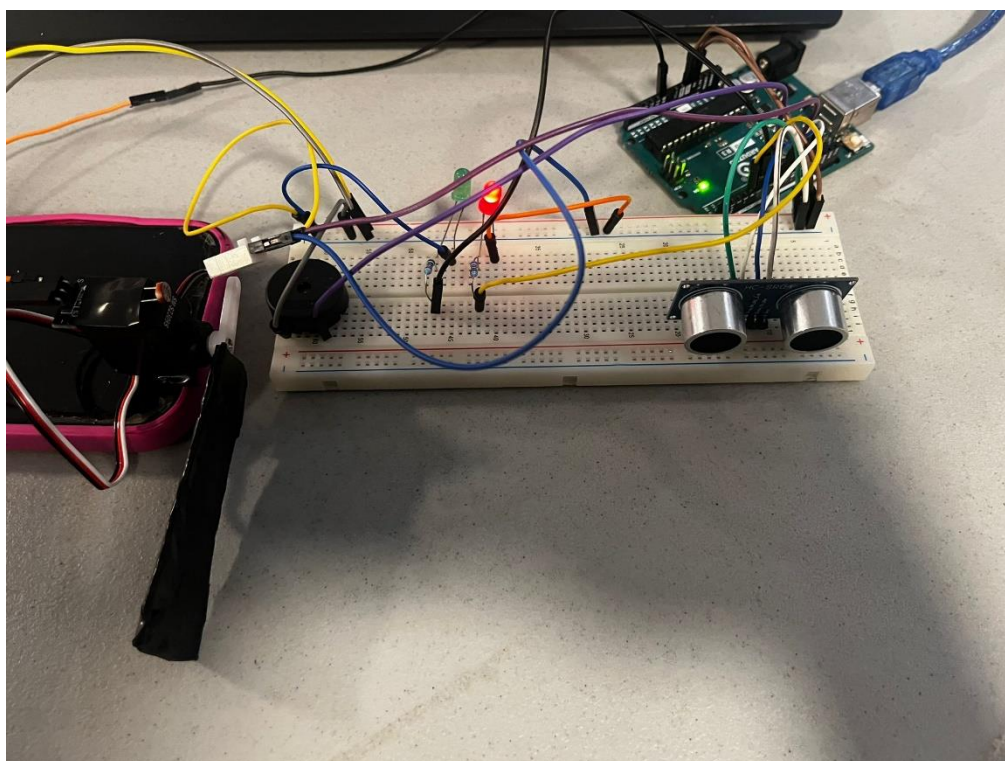


Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

```
}  
  
delay(500);  
  
}
```

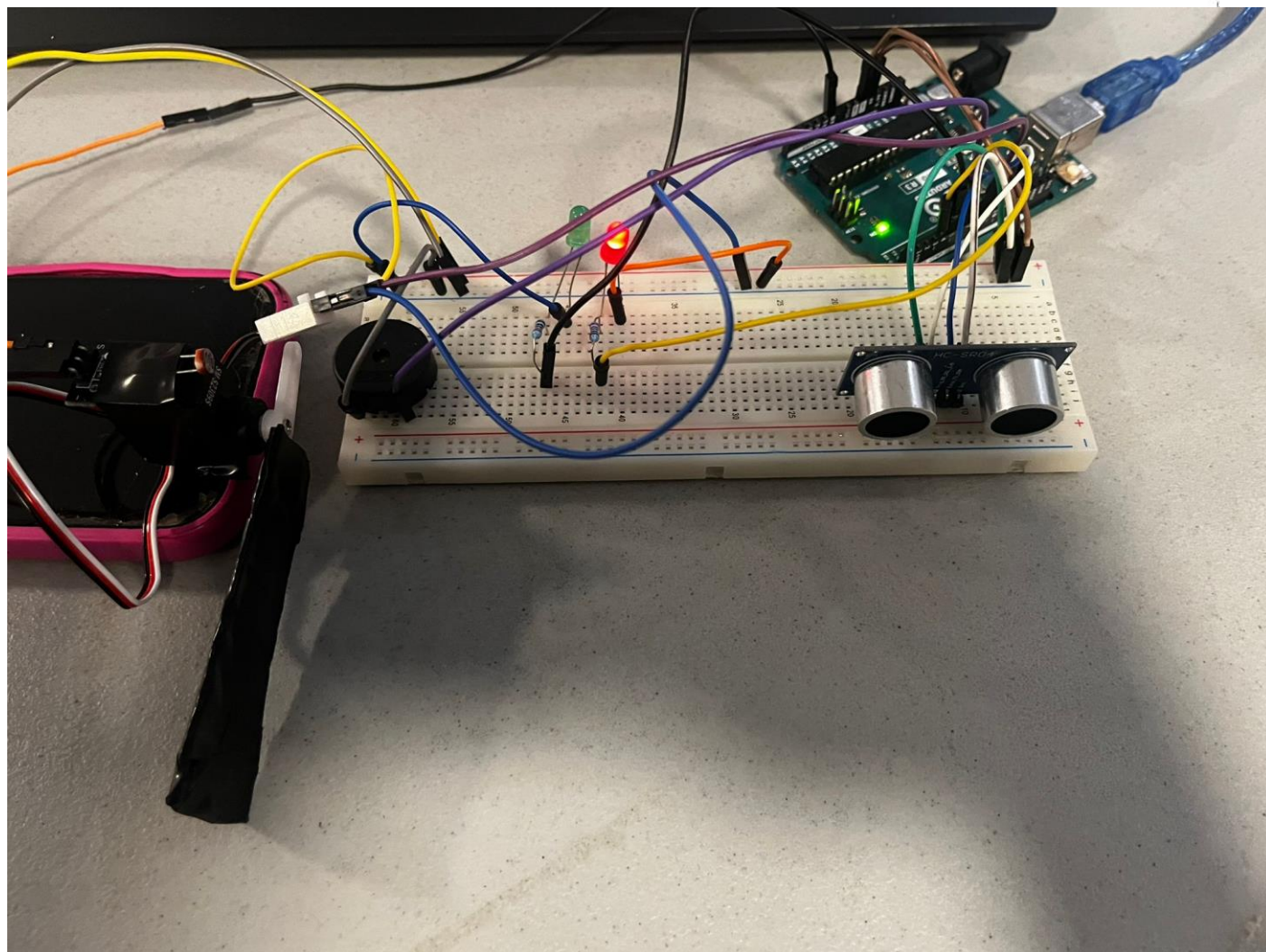




Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573





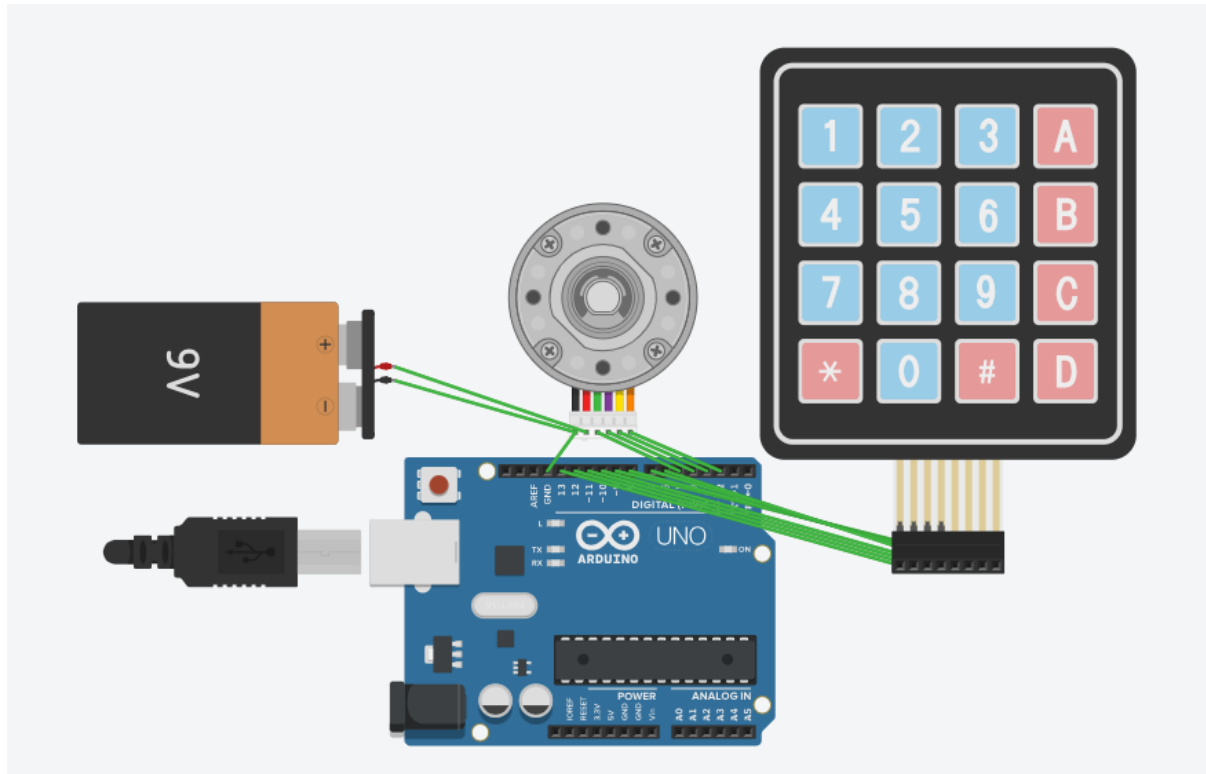
Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

PROIECT: Conveyor Stepper

CODE MOTOR+KEYPAD+BUTON



This Arduino Uno circuit controls a stepper motor using a 4x4 keypad. The system is powered externally by a 9V battery. When you press a number, the Arduino reads the input. It immediately calculates the steps needed for that distance. The motor then rotates to move exactly that many centimeters.

Component Overview

- Arduino Uno: The main microcontroller brain.
- 4x4 Keypad: Inputs the desired distance values (in centimeters).
- Stepper Motor: Executes rotational movements.

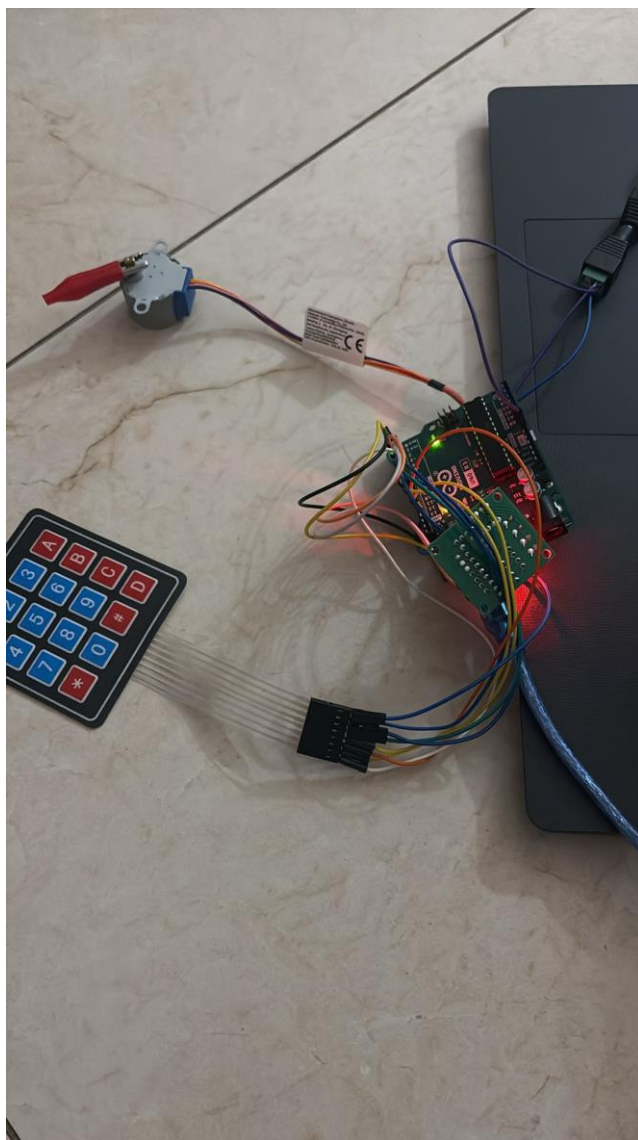


Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

- 9V Battery: Provides external power to the motor.

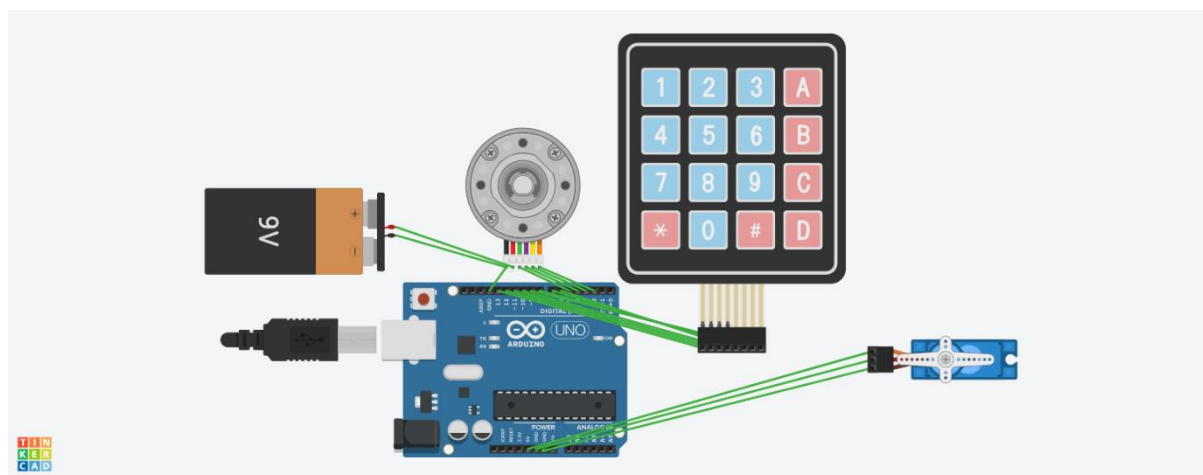


PROIECT FINAL: Car Presentation Showroom

Descriere: Un sistem care prezintă într-un garaj un model de mașină, pe o platformă care se rotește prin introducerea unei valori prin Keypad.

Objective of the Motor System

The objective of this motor-based automatic parking system is to automate the movement of a car and the opening/closing of the parking barrier. The stepper motor rotates the platform to simulate the car leaving the parking area, while the servo motor controls the barrier gate automatically.



Main Components

- Arduino board
- Stepper motor
- Servo motor
- Cardboard rotating platform
- Barrier gate mechanism
- Power supply
- Connecting wires and electronic components



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

How It Works

1. Arduino sends commands to the stepper motor.
2. The stepper motor rotates the platform with the car.
3. When the car reaches the exit, the servo motor lifts the barrier.
4. After the car passes, the barrier closes automatically.

Advantages

- Automates the parking exit system.
- Low power consumption.
- Demonstrates the use of different types of motors.
- Can be expanded with sensors and displays.
- Helps students learn programming and electronics.
- Low-cost project.

Disadvantages

- Motor speed is limited.
- Requires stable power supply.
- The cardboard structure is not very durable.
- The movement may become inaccurate if the stepper motor loses steps.
- Requires programming and wiring knowledge.

Applications

- Automatic parking systems.



Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+ 2025-1-RO01-KA122-VET-000340573

- Entrance and exit barriers.
- Smart city systems.
- Educational robotics projects.
- Industrial automation simulations.
- New released cars presentation

CODE:

```
#include <Keypad.h>
#include <Stepper.h>
#include <Servo.h>
int led1=A1;
int led2=A2;
const int STEPS_PER_REV = 32;
Stepper motor(STEPS_PER_REV, 2, 4, 3, 5);
Servo motoras;
float dstmm = 51.2;
long nr = 0;
int directie = 1;
const byte ROW = 4;
const byte COL = 4;
char matricenumere[ROW][COL] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

byte RowPins[ROW] = {10, 11, 12, 13};
byte ColPins[COL] = {6, 7, 8, 9};

Keypad customKeypad = Keypad(makeKeymap(matricenumere), RowPins, ColPins, ROW, COL);

int ok=0;

void setup() {
    Serial.begin(9600);
```



Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+ 2025-1-RO01-KA122-VET-000340573

```

motor.setSpeed(250);
Serial.println("Sistem pregatit.");
motoras.attach(A0);
motoras.write(32);
pinMode(led1,OUTPUT);
pinMode(led2,OUTPUT);
}

void loop() {
    char customKey = customKeypad.getKey();
    int cnt=0;
    if (customKey) {
        if (customKey >= '0' && customKey <= '9') {
            nr = nr * 10 + (customKey - '0');
            Serial.print("Numar in formare: ");
            Serial.println(customKey);
        }

        else if (customKey == '#') {
            Serial.print("Numar final: ");
            Serial.println(nr);
            directie=1;
            long stepsReq = nr * dstmm;
            Serial.println("Start. A=Pausa || B=Reverse || C=Stop");

            for (long i = 0; i < stepsReq; i++) {
                motor.step(1 * directie);

                char liveKey = customKeypad.getKey();

                // --- LOGICA PAUZA (Tasta A) ---
                if (liveKey == 'A') {
                    Serial.println("PAUZA. Apasa A pentru a continua...");
                    delay(500); // Evita citirea dubla a tastei

                    bool inPauza = true;
                    while(inPauza) {
                        char pauseKey = customKeypad.getKey();
                        if(pauseKey == 'A') {
                            Serial.println("RESUME!");
                            inPauza = false;
                        }
                    }
                }
            }
        }
    }
}

```



Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+ 2025-1-RO01-KA122-VET-000340573

```
        delay(500); // Evita reluarea imediata a pauzei
    }
}

// --- LOGICA REVERSE (Tasta B) ---
if (liveKey == 'B') {
    directie *= -1;
    Serial.println("Directie schimbata!");
}

// --- LOGICA OPRIRE (Tasta C) ---
if (liveKey == 'C'){
    Serial.println("Cursa oprita fortat! Introdu alt numar.");
    directie=0;
    ok=1;
    nr=0;
}

if (liveKey == 'D') {
    cnt++;
    Serial.print("Contor: ");
    Serial.println(cnt);
}

if(cnt%2!=0){
    motoras.write(125);
    analogWrite(led1,255);
    analogWrite(led2,255);
}
else {
    motoras.write(32);
    analogWrite(led1,0);
    analogWrite(led2,0);
}
}

if(ok==0){
    Serial.println("Cursa terminata. Introdu alt numar.");
    nr = 0;
}
}
```



Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

```
else if (customKey == '*') {  
    nr = 0;  
    Serial.println("Resetat. Introdu alt numar.");  
}  
}  
}
```





Cofinanțat de
Uniunea Europeană



Start în Robotică cu Erasmus+
2025-1-RO01-KA122-VET-000340573

