



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

PORTOFOLIU DE PRACTICĂ

MOBILITATE CIPRU

10.05.2026 – 23.05.2026

MĂZĂREANU TEODORA-ELENA

SĂLBATECU ARIS

STOIAN DANIEL



Cofinanțat de
Uniunea Europeană



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

Cuprins

PROIECT: Traffic Light	2
PROIECT: Fading	3
PROIECT: Laser	4
PROIECT: Tilt.....	5
PROIECT: Tilt Alarm System	6
PROIECT: Arduino Motor	11
PROIECT: Stepper motor system.....	14
PROIECT FINAL: Sistem de nivelare	17



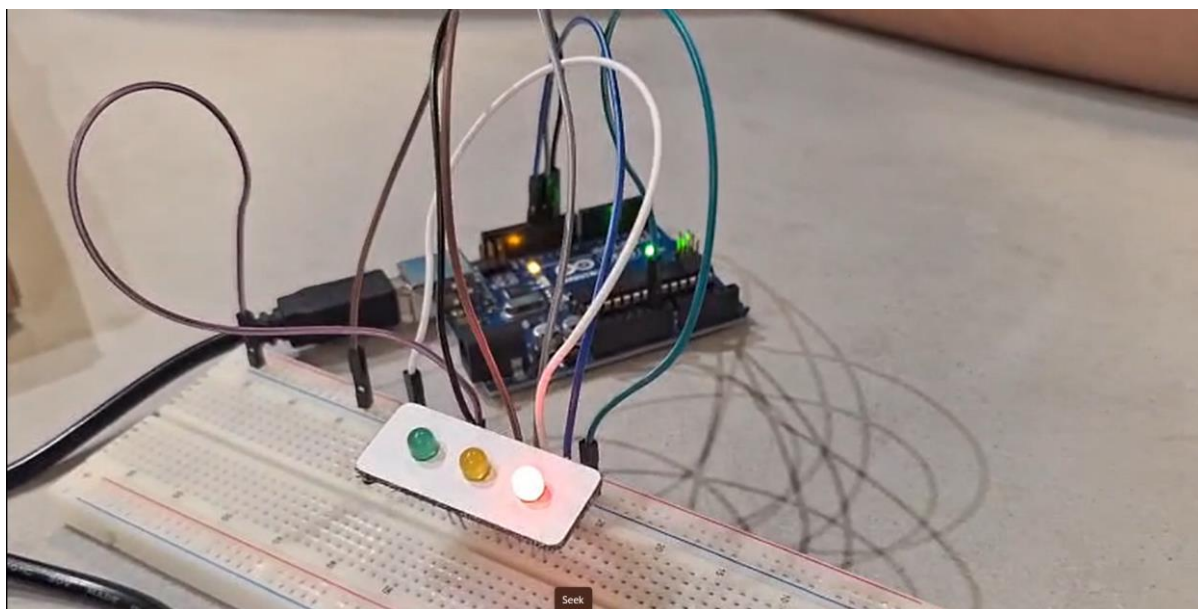
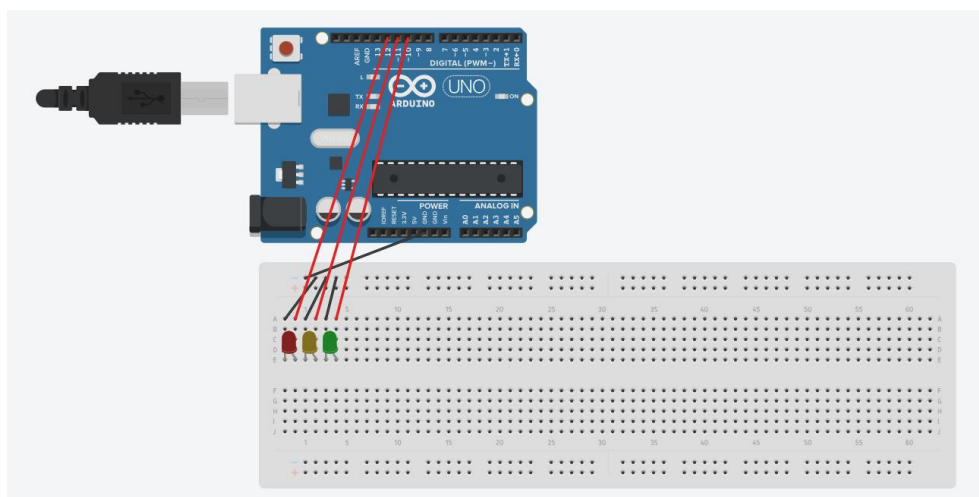
Cofinanțat de
Uniunea Europeană



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

PROIECT: Traffic Light

[Link material video](#)





Cofinanțat de
Uniunea Europeană



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

PROIECT: Fading

CODE:

```
int pinled=9;

void setup() {

    Serial.begin(9600);

}

void loop() {

    int sensorValue = analogRead(A0);

    int brightness = sensorValue * (255.0/1023.0);

    analogWrite(pinled, brightness);

    Serial.println(brightness);

}
```



Cofinanțat de
Uniunea Europeană



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

PROIECT: Laser

Code:

```
const int laserPin=13;

void setup() {

    pinMode(laserPin, OUTPUT);

}

void loop() {

    digitalWrite(laserPin, HIGH); // Turn on the laser head

    delay(1000); // wait for one second

    digitalWrite(laserPin, LOW); // Turn off the laser head

    delay(1000);

}
```



Cofinanțat de
Uniunea Europeană



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

PROIECT: Tilt

CODE:

```
int ledPin = 11;

int inPin = 7;

int value = 0;

void setup()

{

    pinMode(ledPin, OUTPUT);           // initializes digital pin 13 as output

    pinMode(inPin, INPUT);             // initializes digital pin 7 as input

}

void loop()

{

    value = digitalRead(inPin);        // reads the value at a digital input

    digitalWrite(ledPin, value);

}
```



Cofinanțat de
Uniunea Europeană



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

PROIECT: Tilt Alarm System

This is a system where there are two possible states: when the tilt sensor is off, the LCD displays "Sistem Normal", the green LED turns on, and the buzzer stays silent. When the tilt sensor is on, the LCD displays "Alert Stabilise", the green LED turns off, the red LED turns on, and the buzzer starts beeping.

LCD Screen:

- VSS (Pin 1) -> GND
- VDD (Pin 2) -> 5V
- V0 (Pin 3) -> GND
- RS (Pin 4) -> Digital Pin 12
- RW (Pin 5) -> GND
- E / Enable (Pin 6) -> Digital Pin 11
- D4, D5, D6, D7 -> Digital Pins 5, 4, 3, 2.
- A / Anode (Pin 15) -> 5V
- K / Cathode (Pin 16) -> GND

Indicators and Sensors:

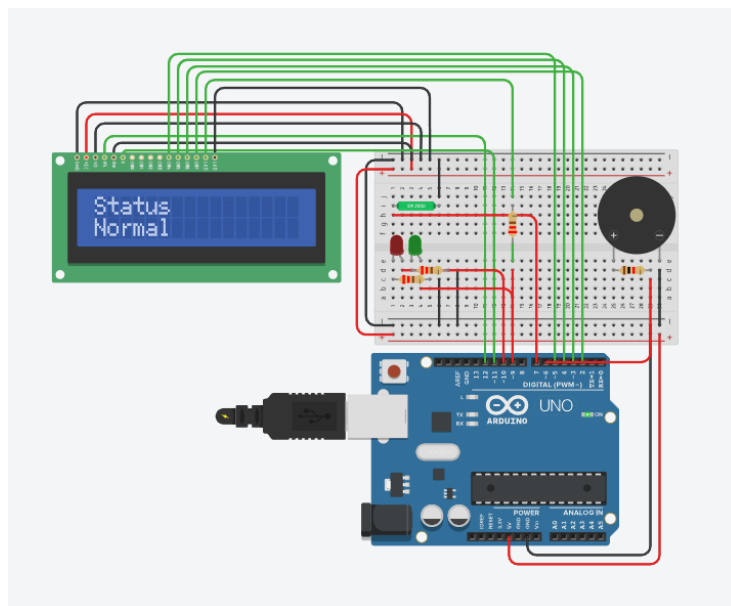
- Green LED: Positive to Pin 9, negative to GND via a resistor.
- Red LED: Positive to Pin 10; negative to GND via a resistor.
- Buzzer: Positive to Pin 6, negative to GND.
- Tilt Sensor: One leg to GND, the other leg to Digital Pin 7.



Cofinanțat de
Uniunea Europeană



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



```
#include <LiquidCrystal.h>
```

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;  
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
const int tiltPin = 7;  
const int redPin = 10;  
const int greenPin = 9;  
const int buzzerPin = 6;
```

```
int value = 0;
```

```
void setup() {  
  pinMode(redPin, OUTPUT);  
  pinMode(greenPin, OUTPUT);  
  pinMode(buzzerPin, OUTPUT);  
  pinMode(tiltPin, INPUT_PULLUP);
```

```
  lcd.begin(16, 2);  
  lcd.print("Hello!!!!");  
  delay(1000);  
  lcd.clear();  
}
```

```
void loop() {
```



Cofinanțat de
Uniunea Europeană



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

```
value = digitalRead(tiltPin);

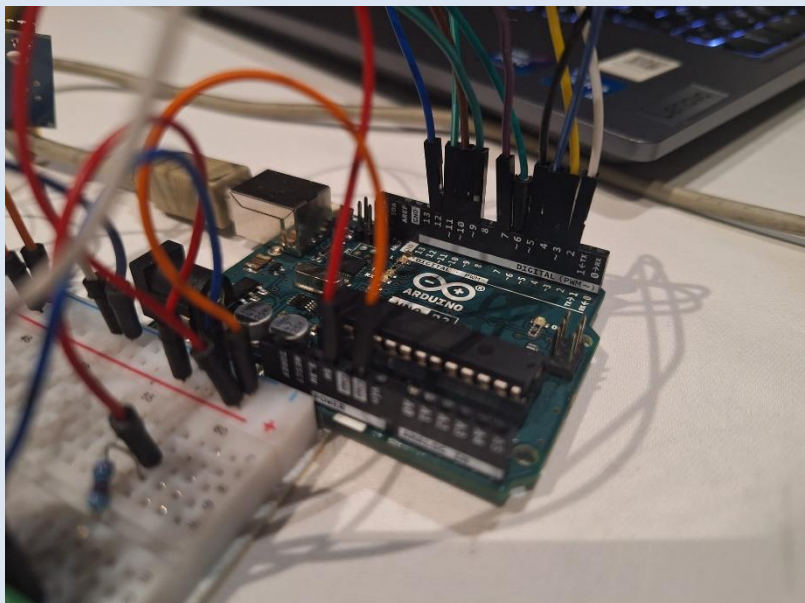
if (value == HIGH) {
  digitalWrite(redPin, HIGH);
  digitalWrite(greenPin, LOW);

  tone(buzzerPin, 1000);

  lcd.setCursor(0, 0);
  lcd.print("Alert      ");
  lcd.setCursor(0, 1);
  lcd.print("Stabilise   ");
}
else {
  digitalWrite(redPin, LOW);
  digitalWrite(greenPin, HIGH);

  noTone(buzzerPin);

  lcd.setCursor(0, 0);
  lcd.print("Status      ");
  lcd.setCursor(0, 1);
  lcd.print("Normal      ");
}
}
```

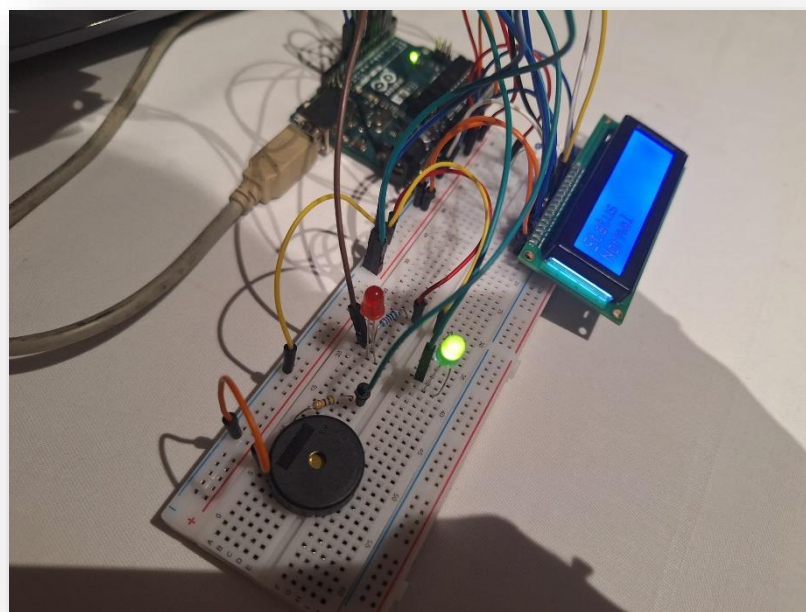
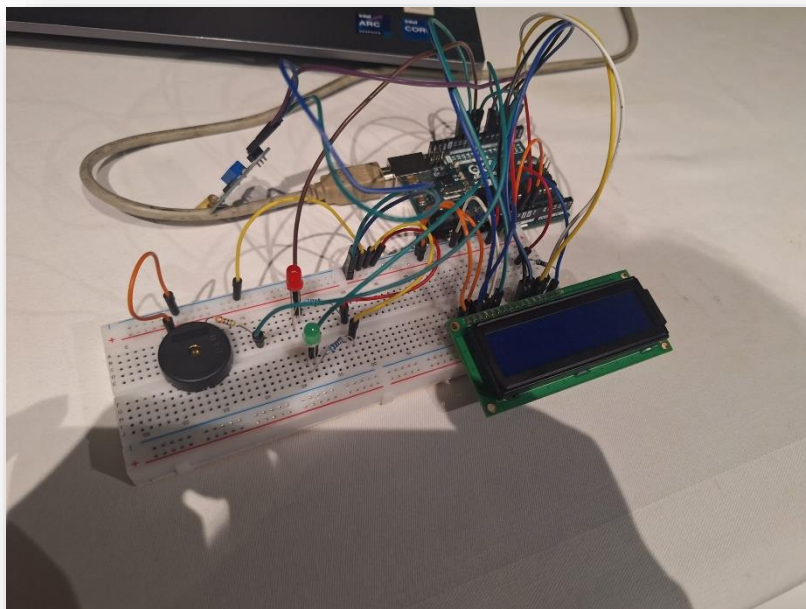




Cofinanțat de
Uniunea Europeană



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

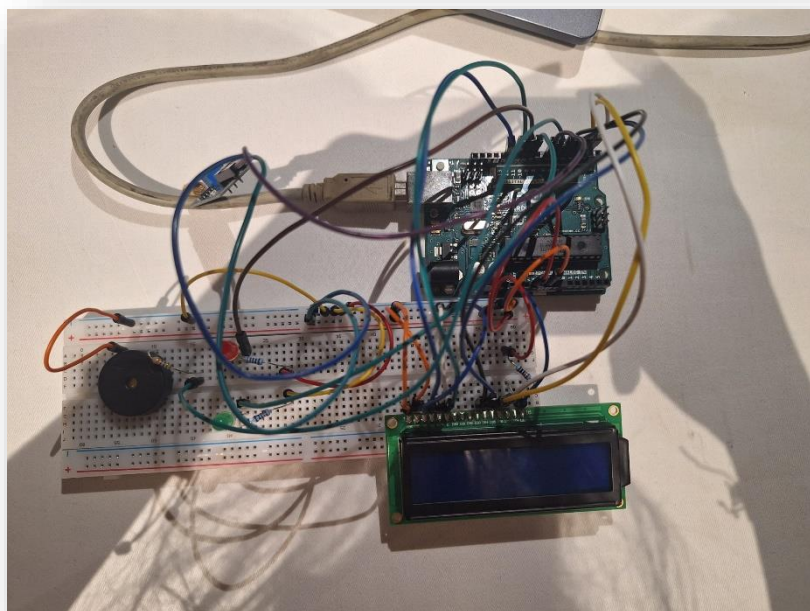
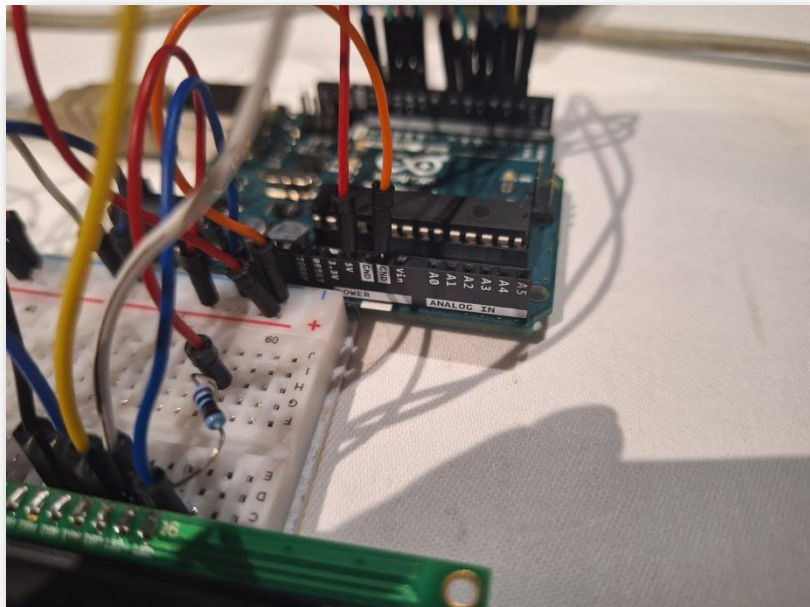




Cofinanțat de
Uniunea Europeană



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





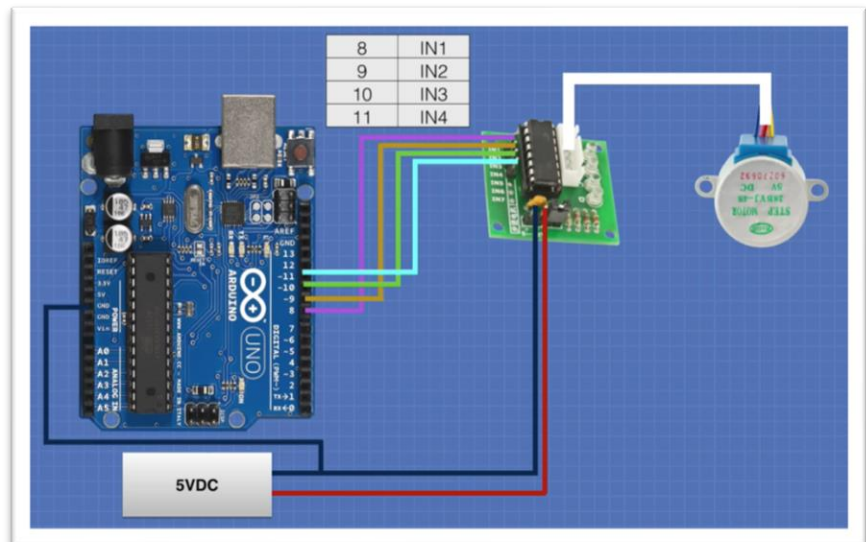
Cofinanțat de
Uniunea Europeană



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

PROIECT: Arduino Motor

[Link material video](#)



CODE :

```
//Include the Arduino Stepper Library

#include <Stepper.h>

// Define Constants

// Number of steps per internal motor revolution

const float STEPS_PER_REV = 32;

// Amount of Gear Reduction
```



Cofinanțat de
Uniunea Europeană



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

```
const float GEAR_RED = 64;

// Number of steps per geared output rotation

const float STEPS_PER_OUT_REV = STEPS_PER_REV * GEAR_RED;

// Define Variables

// Number of Steps Required

int StepsRequired;

// Create Instance of Stepper Class

// Specify Pins used for motor coils

// The pins used are 8,9,10,11

// Connected to ULN2003 Motor Driver In1, In2, In3, In4

// Pins entered in sequence 1-3-2-4 for proper step sequencing

Stepper steppermotor(STEPS_PER_REV, 8, 10, 9, 11);

void setup()
{
    // Nothing (Stepper Library sets pins as outputs)
}

void loop()
```



Cofinanțat de
Uniunea Europeană



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

```
{  
  
    // Slow - 4-step CW sequence to observe lights on driver board  
  
    steppermotor.setSpeed(1);  
  
    StepsRequired  =  4;  
  
    steppermotor.step(StepsRequired);  
  
    delay(2000);  
  
  
    // Rotate CW 1/2 turn slowly  
  
    StepsRequired  =  STEPS_PER_OUT_REV / 2;  
  
    steppermotor.setSpeed(100);  
  
    steppermotor.step(StepsRequired);  
  
    delay(1000);  
  
  
    // Rotate CCW 1/2 turn quickly  
  
    StepsRequired  =  - STEPS_PER_OUT_REV / 2;  
  
    steppermotor.setSpeed(700);  
  
    steppermotor.step(StepsRequired);  
  
    delay(2000);  
  
}
```



Cofinanțat de
Uniunea Europeană



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

PROIECT: Stepper motor system

[Link material video](#)

This is a stepper motor system where, based on an input from a keypad, the stepper motor will do the number of revolutions necessary to move a cube placed on a screw for the same number of millimeters as the input from the keypad. (A full stepper motor revolution is 2mm, if the keypad input is 20, the stepper motor will do 10 revolutions)

```
#include <Keypad.h>
#include <Stepper.h>

const int button = A2;
int buttonstate=0;

const float STEPS_PER_REV = 32;
const float GEAR_RED = 64;
const float STEPS_PER_OUT_REV = STEPS_PER_REV * GEAR_RED;

long StepsRequired;
long inputValue = 0;

Stepper steppermotor(STEPS_PER_REV, 8, 10, 9, 11);

const byte ROWS = 4;
const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {13, 12, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

void setup() {
  Serial.begin(9600);
  Serial.println("Enter a number and press # to confirm.");
```



Cofinanțat de
Uniunea Europeană



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

```
pinMode(button, INPUT);
}

void loop() {
  char key = customKeypad.getKey();
  if (key) {
    if (key >= '0' && key <= '9') {
      inputValue = (inputValue * 10) + (key - '0');
      Serial.print("Typing (mm): ");
      Serial.println(inputValue);
    }
    else if (key == '#') {
      Serial.print("Confirmed target: ");
      Serial.print(inputValue);
      Serial.println(" mm");
      float stepsPerMM = STEPS_PER_OUT_REV / 2.0;
      StepsRequired = inputValue * stepsPerMM;
      Serial.print("Moving motor ");
      Serial.print(StepsRequired);
      Serial.println(" steps.");
      steppermotor.setSpeed(1000);
      steppermotor.step(StepsRequired);
      inputValue = 0;
      buttonstate=digitalRead(button);
      if(buttonstate == HIGH)
      {
        StepsRequired = -1 * StepsRequired;
        steppermotor.step(StepsRequired);
        Serial.println("Button pressed. Going back to initial state.");
      }
      else
        Serial.println("Motor finished. Enter a new distance in mm:");
    }
  }
}
```



Cofinanțat de
Uniunea Europeană



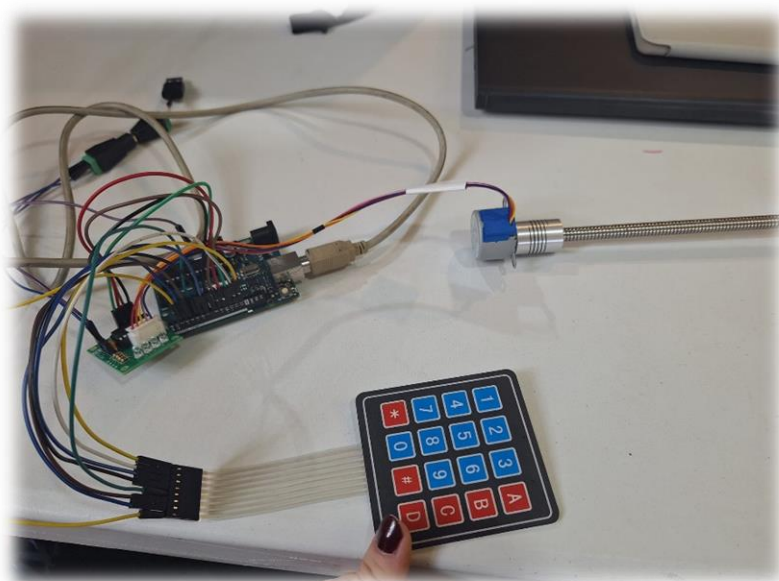
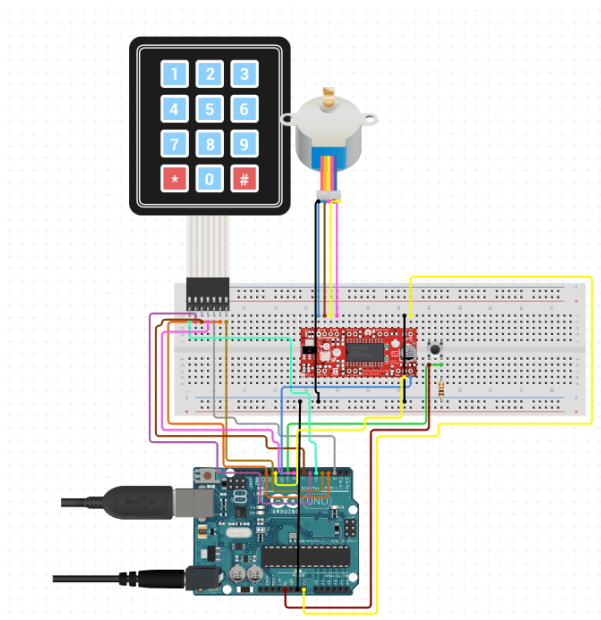
Start în Robotică cu Erasmus+ 2025-1-RO01-KA122-VET-000340573

Usages:

- The Foundation of a CNC Machine or 3D Printer

This represents one of the axes on most 3D printers and CNC routers.

- Automated Feeder or Sorter
- Smart Home Automation





Cofinanțat de
Uniunea Europeană



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

PROIECT FINAL: Sistem de nivelare

Descriere: Un sistem care monitorizează unghiul de înclinare a unui obiect și atenționează când acesta nu este drept.

[Link material video](#)

This is a stepper motor system where, based on an input from a keypad, the stepper motor will do the number of revolutions necessary to move a cube placed on a screw for the same number of millimeters as the input from the keypad (A full stepper motor revolution is 2mm, if the keypad input is 20, the stepper motor will do 10 revolutions). There is a tilt sensor, that when it is activated, the stepper motor will stop working and a button that stops the motor when it is engaged.

Components:

- Arduino board;
- Keypad;
- 2 LEDs;
- Tilt sensor;
- Buzzer;
- Stepper motor;
- Breadboard;
- Button.

Code:

```
#include <Keypad.h>

#include <Stepper.h>

const int button = A1;

const float STEPS_PER_REV = 32;

const float GEAR_RED = 64;

const float STEPS_PER_OUT_REV = STEPS_PER_REV * GEAR_RED;

const int buzzer = A2;

const int ledstop = A3;

const int ledmerge = A4;

const int tilt = A5;
```



Cofinanțat de
Uniunea Europeană



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

```
long StepsRequired;

long inputValue = 0;

Stepper steppermotor(STEPS_PER_REV, 8, 10, 9, 11);

const byte ROWS = 4;
const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {2, 3, 4, 5};
byte colPins[COLS] = {6, 7, 12, 13};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

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

    Serial.println("Enter a number and press # to confirm.");

    pinMode(button, INPUT);
```



Cofinanțat de
Uniunea Europeană



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

```
pinMode(ledstop, OUTPUT);

pinMode(ledmerge, OUTPUT);

pinMode(buzzer, OUTPUT);

pinMode(tilt, INPUT);

}

void loop() {

    char key = customKeypad.getKey();

    if (key) {

        if (key >= '0' && key <= '9') {

            inputValue = (inputValue * 10) + (key - '0');

            Serial.print("Typing (mm): ");

            Serial.println(inputValue);

        }

        else if (key == '#') {

            Serial.print("Confirmed target: ");

            Serial.print(inputValue);

            Serial.println(" mm");

            digitalWrite(ledmerge, HIGH);

            digitalWrite(ledstop, LOW);

            tone(buzzer, 5);

            delay(1000);

            noTone(buzzer);

            float stepsPerMM = STEPS_PER_OUT_REV / 2.0;

            StepsRequired = inputValue * stepsPerMM;
```



Cofinanțat de
Uniunea Europeană



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

```
Serial.print("Moving motor ");

Serial.print(StepsRequired);

Serial.println(" steps.");

steppermotor.setSpeed(1000);


bool stopped = false;

for (long i = 0; i < StepsRequired; i++) {

    steppermotor.step(1);

    if (digitalRead(button) == HIGH || digitalRead(tilt) == HIGH) {

        if (digitalRead(button) == HIGH) {

            Serial.println("Button pressed. Stopping.");

        } else {

            Serial.println("Tilt detected! Stopping.");

        }

        digitalWrite(ledstop, HIGH);

        digitalWrite(ledmerge, LOW);

        tone(buzzer, 5);

        delay(1000);

        noTone(buzzer);

        stopped = true;

        break;

    }

}

inputValue = 0;
```



Cofinanțat de
Uniunea Europeană



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

```
if (!stopped) {  
    Serial.println("Motor finished. Enter a new distance in mm:");  
    digitalWrite(ledstop, HIGH);  
    digitalWrite(ledmerge, LOW);  
    tone(buzzer, 5);  
    delay(1000);  
    noTone(buzzer);  
}  
}  
}  
}  
}
```

Usages:

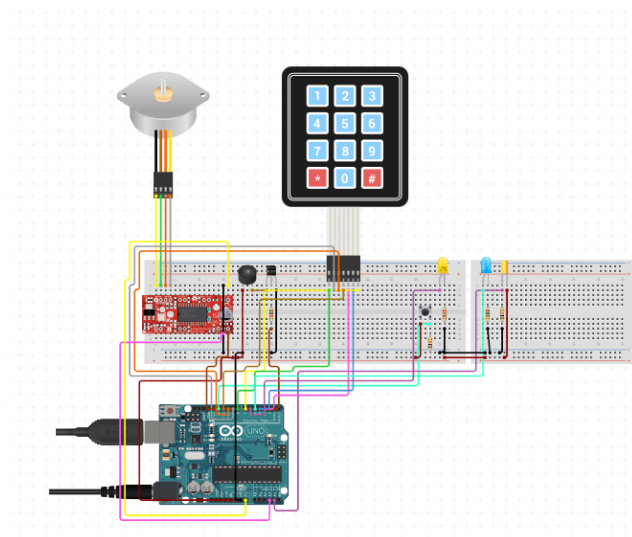
- Checking the mass limit for lifts.
- Automated Leveling Systems
- Start of a CNC Router



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