Lift în miniatură

# Membrii echipei

1. Cîrloban Vlad-Petrișor
2. Cosmulete Ion-Cosmin
3. Șova Ioan-Rareș

# Contribuții în cadrul proiectului

* Cîrloban Vlad-Petrișor a comandat piesele necesare pentru realizarea proiectului: motorul pas cu pas, scripetele, senzorul ultrasonic.
* Cosmulete Ion-Cosmin a realizat acționarea liftului, cu ajutorul motorului pas cu pas, și logica de funcționare a butoanelor și a senzorului infraroșu controlat de telecomandă.
* Șova Ioan-Rareș, alături de Cîrloban Vlad-Petrișor, a conectat senzorul ultrasonic și a implementat afișarea etajelor pe ecran.
* Toți membrii echipei au contribuit la construirea machetei de lift.

# Obiectivele proiectului

Proiectul nostru vizează studierea unui sistem de acționare electrică pentru un lift, care simulează funcționarea unui motor pas cu pas, precum și implementarea unui sistem de comandă pentru poziționarea liftului.

Totodată, prin această lucrare ne propunem să aplicăm cunoștințele dobândite despre mașini electrice, și anume: tipuri de motoare electrice, surse de alimentare, controlul turației și al sensului de rotație, senzori de poziție, microcontrolere.

# Descrierea soluției implementate cu prezentarea funcționalităților aferente soluției

## Arhitectura hardware

Sistemul de lift este compus dintr-o structură hardware modulară, care integrează elemente de acționare, control, detecție și interfață cu utilizatorul. Arhitectura este proiectată astfel încât să asigure funcționarea manuală a liftului.

Pentru acționarea mecanismului de ridicare/coborâre este folosit un **motor pas cu pas 28BYJ-48** alimentat la 9V, simulat în mediul Tinkercad. Motorul este cuplat mecanic la un scripete, prin care este fixată o ață ce acționează cabina liftului. De asemenea, există încă o ață care asigură stabilitatea cabinei. Acest sistem permite transmiterea forței într-un mod simplu, dar eficient.

Motorul pas cu pas este comandat prin intermediul unui **driver de tip ULN2003**. Acesta primește semnale de tip PWM (modulare a lățimii impulsurilor) și semnale logice de la microcontrolerul principal (Arduino UNO), controlând secvența de activare a înfășurărilor pentru a genera mișcarea de rotație. Driverul permite și inversarea sensului de rotație, în funcție de cerința de urcare sau coborâre a liftului.

Proiectul utilizează **două plăci de dezvoltare Arduino UNO**, fiecare având funcții distincte:

* Unul dintre microcontrolere este responsabil de citirea distanței de la senzorul ultrasonic și afișarea pe ecran.
* Celălalt controlează logica de mișcare, incluzând acționarea motorului.

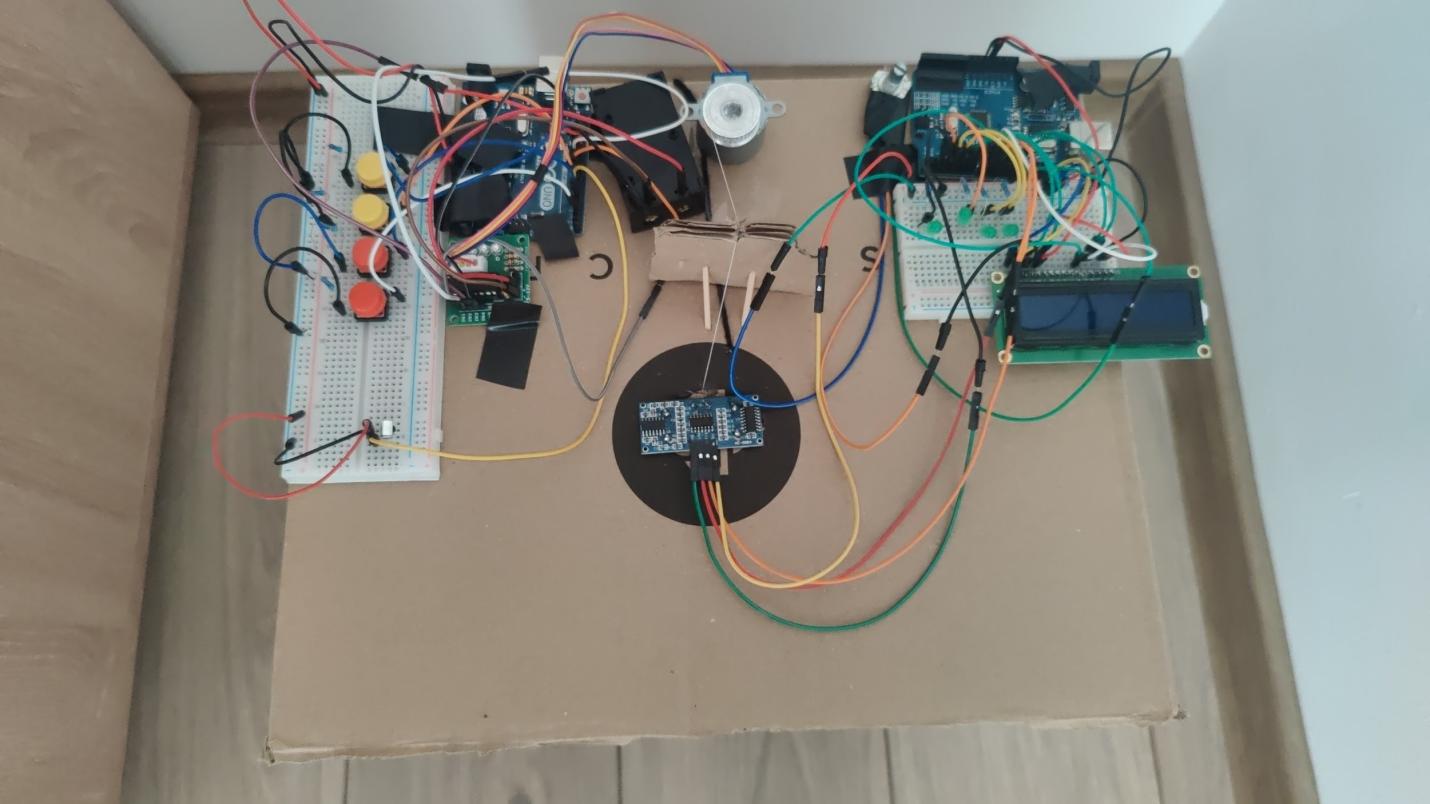
Această separare permite o arhitectură mai clară, cu sarcini împărțite eficient între cele două unități.

Un **senzor ultrasonic HC-SR04** este utilizat pentru detecția poziției liftului față de bază, fără contact fizic. Acesta calculează distanța până la cabina liftului și permite microcontrolerului să determine etajul curent.

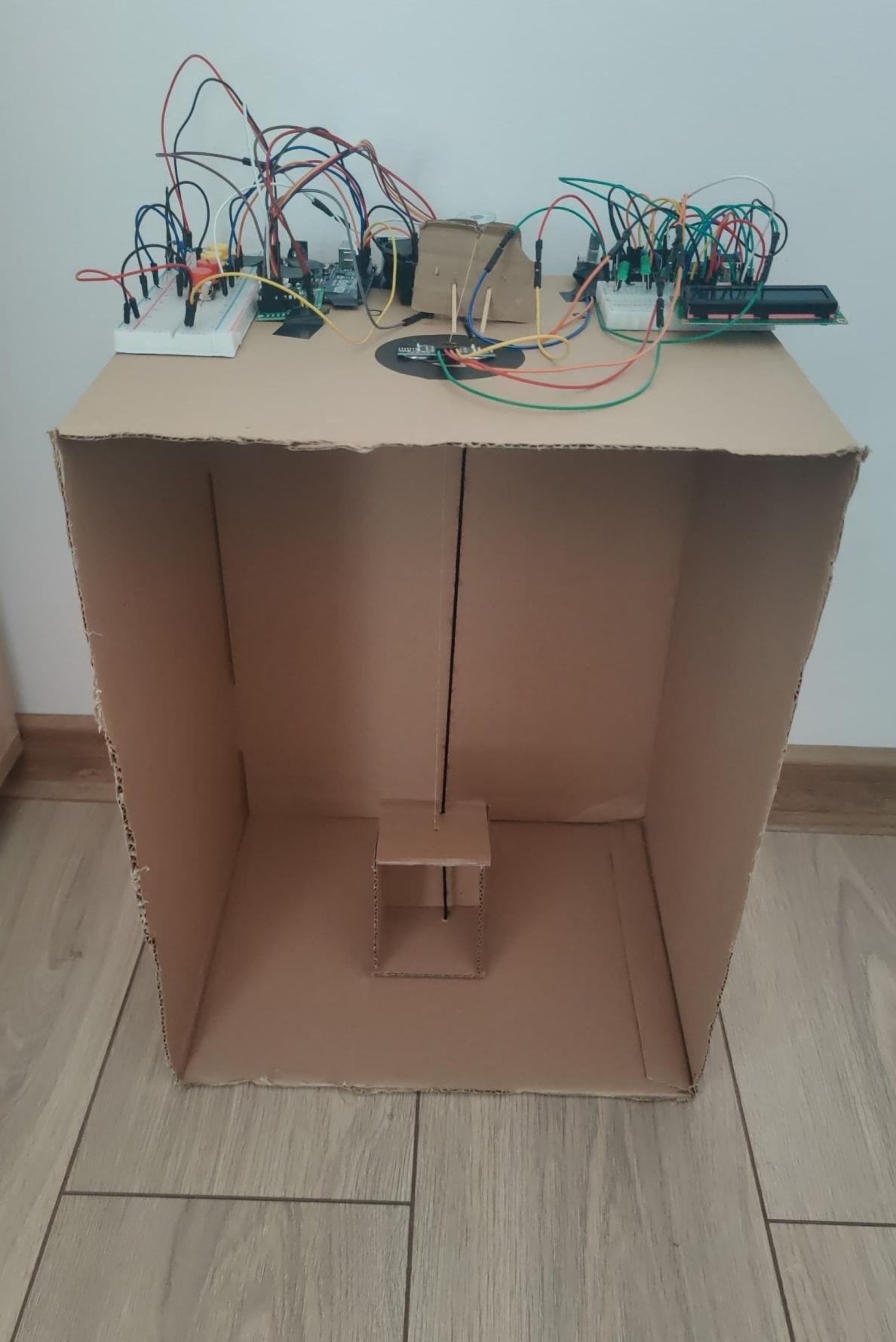
Pentru preluarea comenzilor de la distanță, este integrat un **senzor infraroșu** compatibil cu o telecomandă uzuală. Acest sistem oferă o modalitate comodă de interacțiune, fără fir, între utilizator și lift.

Fiecare etaj este asociat cu un **buton fizic** dedicat, care permite selecția manuală a etajului dorit. În plus, sunt montate **LED-uri de stare** care indică în mod vizual etajul la care se află liftul în orice moment.

Un **ecran LCD 16:2 cu microcontroler HD44780** este folosit pentru afișarea etajului curent. Contrastul afișajului este reglat printr-un **potențiometru**.



***Figura 1:*** *Evidențierea pieselor*



***Figura 2:*** *Macheta de lift*

## Arhitectura software

Arhitectura software este proiectată pentru a asigura o interacțiune fluidă între componentele hardware și utilizator. Programul implementat pe plăcile Arduino permite controlul manual al liftului, astfel încât utilizatorul selectează etajul dorit prin apăsarea unui buton fizic sau a unuia dintre butoanele telecomenzii. Microcontrolerul interpretează comanda, determină direcția de mișcare și oprește automat liftul atunci când ajunge la etajul ales.

Poziția cabinei liftului este determinată cu ajutorul senzorului ultrasonic. Acesta măsoară în mod continuu distanța față de baza structurii, iar software-ul compară valorile cu praguri definite pentru fiecare etaj. Astfel, se afișează etajul corespunzător.

Programul implementat pe microcontrolerul responsabil de citirea distanței de la senzorul ultrasonic este următorul:

#include <LiquidCrystal.h>

long duration;

int distance;

const int trigPin = 10;

const int echoPin = 11;

const int led1 = 2;

const int led2 = 3;

const int led3 = 4;

const int led4 = 5;

const int rs = 6, en = 7, d4 = 8, d5 = 9, d6 = 12, d7 = 13;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int previousLevel = -1;

void setup() {

lcd.begin(16, 2);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3, OUTPUT);

pinMode(led4, OUTPUT);

Serial.begin(9600);

}

void loop() {

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2;

Serial.print("Distance: ");

Serial.println(distance);

int currentLevel;

if (distance <= 7) {

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

digitalWrite(led3, LOW);

digitalWrite(led4, HIGH);

currentLevel = 4;

} else if (distance <= 15) {

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

digitalWrite(led3, HIGH);

digitalWrite(led4, LOW);

currentLevel = 3;

} else if (distance <= 23) {

digitalWrite(led1, LOW);

digitalWrite(led2, HIGH);

digitalWrite(led3, LOW);

digitalWrite(led4, LOW);

currentLevel = 2;

} else if (distance <= 31) {

digitalWrite(led1, HIGH);

digitalWrite(led2, LOW);

digitalWrite(led3, LOW);

digitalWrite(led4, LOW);

currentLevel = 1;

}

if (currentLevel != previousLevel) {

lcd.clear();

lcd.print("Floor: ");

if (currentLevel == 1) {

lcd.print("P");

} else {

lcd.print(currentLevel - 1);

}

previousLevel = currentLevel;

}

delay(500);

}

Programul care controlează logica de mișcare și semnalizarea este:

#include <AccelStepper.h>

#include <IRremote.h>

const int button1 = 2;

const int button2 = 4;

const int button3 = 7;

const int button4 = 12;

const int rcvPin=5;

IRrecv irrecv(rcvPin);

decode\_results results;

bool isMoving = false;

const int stepsPerFloor = 3500;

AccelStepper stepper(AccelStepper::FULL4WIRE, 8, 10, 9, 11);

int position = 1;

void setup() {

pinMode(button1, INPUT\_PULLUP);

pinMode(button2, INPUT\_PULLUP);

pinMode(button3, INPUT\_PULLUP);

pinMode(button4, INPUT\_PULLUP);

irrecv.enableIRIn();

stepper.setMaxSpeed(700);

stepper.setAcceleration(200);

Serial.begin(9600);

}

void moveToFloor(int targetFloor) {

if (!isMoving && targetFloor != position) {

isMoving = true;

Serial.println(targetFloor);

int stepTarget = (targetFloor - 1) \* stepsPerFloor;

stepper.moveTo(stepTarget);

while (stepper.distanceToGo() != 0) {

stepper.run();

}

position = targetFloor;

isMoving = false;

}

}

void loop() {

if (!isMoving) {

if(IrReceiver.decode()) {

auto value= IrReceiver.decodedIRData.decodedRawData;

switch(value) {

case 4077715200:

Serial.println("1");

moveToFloor(1);

break;

case 3877175040:

Serial.println("2");

moveToFloor(2);

break;

case 2707357440:

Serial.println("3");

moveToFloor(3);

break;

case 4144561920:

Serial.println("4");

moveToFloor(4);

break;

default: Serial.println(value);

}

IrReceiver.resume();

}

if (digitalRead(button1) == LOW) moveToFloor(1);

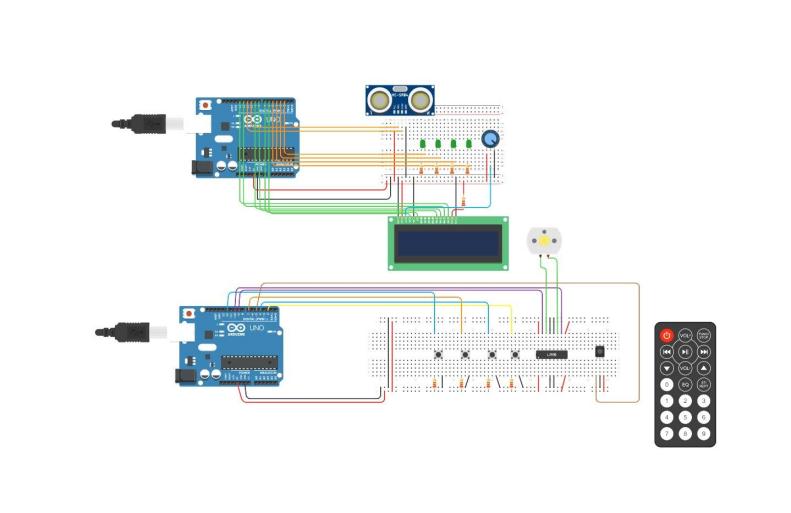
else if (digitalRead(button2) == LOW) moveToFloor(2);

else if (digitalRead(button3) == LOW) moveToFloor(3);

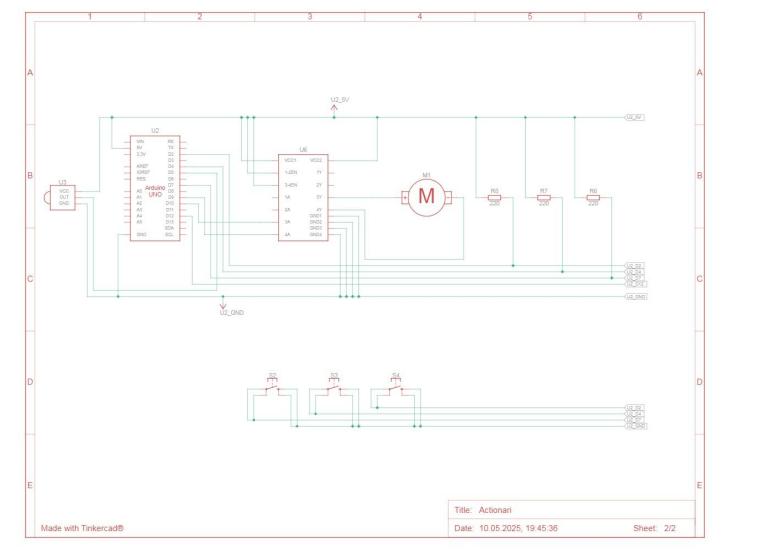
else if (digitalRead(button4) == LOW) moveToFloor(4);

}

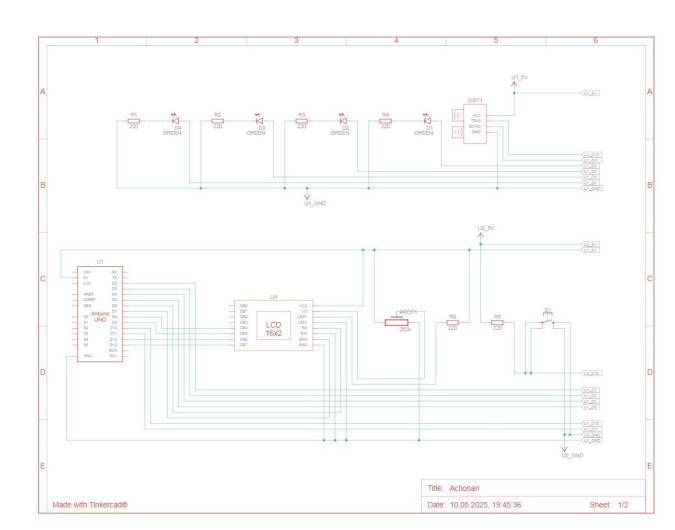
}



***Figura 3:*** *Schema proiectului realizată în Tinkercad*



***Figura 4:*** *Schema logică a acționării efective a liftului*



***Figura 5:*** *Schema logică a interfeței cu utilizatorul*

# Testarea soluției

Soluția noastră a demonstrat posibilitatea realizării unui sistem mecatronic funcțional, utilizând componente accesibile și unelte de simulare educaționale precum Tinkercad. Prin integrarea controlului cu microcontroler, acționării electromecanice și interfeței cu utilizatorul, s-a reușit reproducerea principiilor de bază ale unui lift real la scară redusă.

Arhitectura aleasă a permis controlul precis al poziției liftului și reacția corectă la comenzi manuale. Testările au validat atât acuratețea detecției prin senzor ultrasonic, cât și fiabilitatea comunicării între componentele hardware.

Proiectul oferă o bază solidă pentru extindere ulterioară, putând fi îmbunătățit prin adăugarea unor funcții precum oprirea de urgență, control prin rețea, sau simularea sarcinii în cabina liftului. Totodată, experiența dobândită în proiectarea, programarea și testarea acestui sistem contribuie la înțelegerea practică a noțiunilor de automatizări, electronică și programare pe sisteme încorporate.