

**Universitatea  
Transilvania  
din Brașov**

**FACULTATEA DE INGINERIE ELECTRICĂ  
ȘI ȘTIINȚA CALCULATOARELOR**

## **PROIECT DE DIPLOMĂ**

**Conducător științific:**

**Conf.dr.Ing. Cocias Tiberiu Teodor**

**Absolvent:**

**Paun Andrei**

**BRAȘOV, 2023**

Departamentul Automatica si Tehnologia Informatiei

Programul de studii: Robotica

*Paun Andrei*

# Roboșah: robot asistent pentru jucat șah

Conducător științific:


Conf.dr.ing. Cocias Tiberiu Teodor

Brașov, 2023

FIȘA PROIECTULUI DE DIPLOMĂ

## FIȘA PROIECTULUI DE DIPLOMĂ

Universitatea Transilvania din Brașov	Lucrare de licență/ proiect de diplomă/ disertație nr. ....
Facultatea de Inginerie Electrică și Știința Calculatoarelor	
Departamentul Automatică și Tehnologia Informației	
Programul de studii <b>ROBOTICĂ</b>	Anul universitar 2022-2023
Candidat <b>Paun I. Andrei Ionut</b>	Promoția vară 2023
Conducător științific <b>Conf.dr.ing. Cocias Tiberiu Teodor</b>	Grupa 4LF891
<b>PROIECT DE DIPLOMĂ</b>	
Titlul lucrării: <b>RoboSah. Robot asistent pentru a juca sah</b>	
Problemele principale tratate: Problemele principale care vor fi tratate: 1. Studiul actual al cercetării 2. Descrierea sistemului 3. Implementarea software 4. Proiectarea și simularea sistemului Concluzii și cercetări viitoare	
Locul și durata practicii: Corp V, Laborator V III 15, 60 ore	
Bibliografie: [1] Macesanu G, Cocias T.T., Grigorescu S.M. , Sisteme de vedere artificială : Îndrumar de laborator, 2013 [2] Massimo Banzi, Getting Started with Arduino (Make: Projects), 2009 [3] Paul ViolaMichael J. Jones, Robust Real-Time Face Detection , 2004	
Aspecte particulare: - Nu sunt	
Primit tema la data de: 03.10.2022	
Data predării lucrării:	
Director departament, Prof. dr. ing. Moraru Sorin Aurel 	Conducător științific, Conf.dr.ing. Cocias Tiberiu Teodor

Candidat, Paun I. Andrei Ionut 

PROIECT DE DIPLOMĂ – VIZE –		
Data vizei	Capitole/ problemele analizate	Semnătura conducătorului științific
14.12.2022	Studiu asupra stadiului actual in domeniul si tema vizata; livrabil: raport	
15.03.2023	Proiectare / Arhitecturi, tehnologii, limbaje de programare, algoritmi de utilizat etc; livrabil: raport, document editat	
26.04.2023	Demonstrator / Proof of Concept / Aplicație; livrabil: prototip	
5.06.2023	Documentație; livrabil: document in formatul de proiect de pe situl ATI	
<b>APRECIEREA ȘI AVIZUL CADRULUI DIDACTIC ÎNDRUMĂTOR</b>		
<p>Lucrarea îndeplinește criteriile de calitate pentru a putea fi susținută. Candidatul a prezentat progres remarcabil de-a lungul întregului semestru. Rezultatele experimentale obținute m-au convins să îmi dau acordul pentru susținerea lucrării de licență în cadrul sesiunii vară 2023</p>		
Data: 05.06.2023	ADMIS pentru susținere/ <del>RESPINS</del>	CONDUCĂTOR ȘTIINȚIFIC Conf.dr.ing. Cocias Tiberiu Teodor
<b>AVIZUL DIRECTORULUI DE DEPARTAMENT</b>		
Data: 15.06.2023	ADMIS pentru susținere/ <del>RESPINS</del>	Director departament Prof. dr. ing. Moraru Sorin Aurel
<b>SUSȚINEREA PROIECTULUI DE DIPLOMĂ</b>		
Sesiunea Vară 2023		
Rezultatul susținerii	PROMOVAT cu media:	
	RESPINS cu refacerea lucrării	
	RESPINS fără refacerea lucrării	
<b>PREȘEDINTE COMISIE</b> Prof. dr. ing. Grigorescu Sorin Mihai		

## Listă de figuri, tabele și coduri sursă

## FIGURI

Figura 1. Turcul sistem mecanic de mutat piese de șah .....	11
Figura 2 Garry Kasparov împotriva lui Deep Blue.....	13
Figura 3 Structură pe capitole .....	17
Figura 4 Poziție inițială a pieselor de șah.....	19
Figura 5 Arbore de căutare.....	22
Figura 6 Reprezentare a imaginii color în calculator[6] .....	24
Figura 7 Mișcare liniară în cod G .....	25
Figura 8 Diagramă bloc.....	27
Figura 9 Modul de translație I.....	28
Figura 10 Mecanism transmitere mișcare modul II .....	29
Figura 11 Privere în interior a modului de translație .....	30
Figura 12 Robot Asamblat.....	31
Figura 13 Scut Motor de tip CNC .....	32
Figura 14 Schemă circuit electric.....	33
Figura 15 Împartirea tablei de șah în câmpuri.....	35
Figura 16 Tabla de șah reprezentată în memorie. ....	36
Figura 17 Matrice cu mutări posibile cal .....	37
Figura 18 Rezultate obținute în urma algoritmului de mediere în lumină albastră .....	39
Figura 19 Detecție în lumină naturală .....	40
Figura 20 Funcția de detecție a unui câmp negru gol .....	41
Figura 21 Generare de cod G pentru o succesiune de mutări .....	42
Figura 22 UGS aplicație de transmis cod G .....	43
Figura 23 sistemul mecanic mutând calul.....	44
Figura 24 Ceas de Șah implementat cu arduino.....	45

## TABELE

Tabel 1 Sarcini Propuse.....	15
Tabel 2 Un joc de tip suma 0.....	21

CNC – Calculator cu comandă numerică (Computer numerical control)

CMYK-cyan magenta galben negru (cyan magenta yellow )

HSV-nuantă,saturatie,valoare (hue saturation value)

UGS – trimitător universal de cod G (universal g code sender)

RGB-roșu verde galben (red green blue)

PLA-acid polilactic (polilactic acid)

CAD-grafică asistată de calculator (computer aided design)

CAM-manufactura asistată de calculator (computer aided manufacturing)

## Cuprins

Listă de figuri, tabele și coduri sursă .....	5
Lista de acronime .....	7
.....	10
1.1 Stadiu actual.....	11
1.2 Motivatie .....	14
1.3 Obiective .....	14
1.4 Tehnologii necesare.....	16
1.5 Structura pe capitole.....	16
.....	18
2.1 Jocul de șah.....	19
2.2 Motoare de șah.....	20
2.3 Teoria Jocurilor .....	21
2.4 StockFish .....	21
2.5 Interpretarea scenei .....	23
2.6 Codul G.....	24
.....	26
3.1 Proiectarea Sistemului Mecatronic .....	26
3.1.1 Modul de translație axa X.....	28
3.1.2 Modul de translație de pe axa Y .....	28
3.1.3 Modul de translație axa Z.....	30
3.2 Proiectarea Sistemului Electric .....	31
3.3 Detectia miscarii pieselor de șah folosind imagini .....	34
3.3.1 Functia de preluare a starii .....	34
3.3.2 Functia de generare a codului G.....	36
4.1 Teste si rezultate obtinute.....	38
4.1.1 Preluarea stării.....	38
4.1.2 Generare de cod G.....	41



4.1.3 Testarea sistemului mecanic .....	42
4.2 Utilitatea temei și direcții viitoare de dezvoltare .....	44
5.Bibliografie.....	48
DECLARAȚIE PRIVIND ORIGINALITATEA .....	50

## 1 INTRODUCERE

---

Descrierea proiectului

Istoric

Obiectivele proiectului

Structura pe capitole

---

În zilele noastre avansul tehnologic din domeniul microcontrorălelor și scăderile de preț permit crearea unor sisteme mecatronice specializate capabile de a rezolva sarcini complexe . În domeniul șahului există diferite abordări pentru roboți capabili de a juca șah,de la sisteme integrate până la roboți industriali de uz general care executa mutării primte de la un motor de șah consacrat,

Proiectul de fata își propune să realizeze un robot care joaca șah,robot capabil să mute piesele cu ajutorul modulelor de translație acționate prin moatoare pas cu pas și

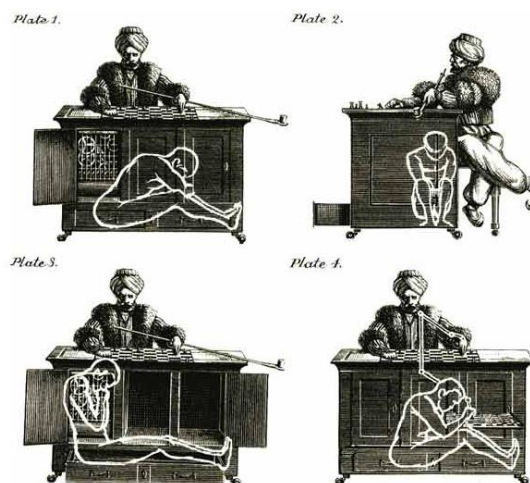
servomotoare, recunoaște situația actuală de pe tabla de șah cu ajutorul unei camere web, procesează informațiile primite cu ajutorul unui calculator, generează o mutare răspuns cu ajutorul unui motor de șah iar mai apoi transmite mutarea către placa Arduino Uno, care prin intermediul unui scut motor de tip CNC execută mutarea robotului.

### 1.1 Stadiu actual

În 1770, Wolfgang von Kempelen a construit un robot cunoscut sub numele de "Turcul", care șah la un nivel foarte ridicat. Împărăteasa Maria Teresa a Austriei și mulți alți oameni au fost impresionați de abilitățile aparente ale acestui robot. Cu toate acestea, în realitate, "Turcul" era o escrocherie elaborată.[1]

Robotul "Turcul" era un sistem mecanic complex, care părea că joacă șah la un nivel înalt și învinge cu ușurință adversari de orice nivel. Însă, în spatele aparențelor, se ascundea un truc ingenios. În interiorul mașinăriei, se afla de fapt un manipulator uman, de obicei un mare maestru al șahului. Acest maestru primea mutările din exteriorul mașinăriei prin intermediul unor mecanisme magnetice și, cu ajutorul unui braț robotic, executa mutările pe tabla de șah atașată ansamblului.

Scopul acestui truc era de a crea iluzia că robotul era capabil să gândească și să joace singur. "Turcul" a fost prezentat în fața unui public numeros și a reușit să înșele mulți oameni cu aparența sa realistă și performanțele sale impresionante.



*Figura 1. Turcul sistem mecanic de mutat piese de șah*

În anii 1900, s-au făcut progrese semnificative în dezvoltarea roboților capabili să joace șah. Un exemplu notabil este robotul construit în Spania în anul 1912 de către Leonardo

Torres y Quevedo. Acesta este considerat primul calculator mecanic capabil să joace jocuri din lume[2].

Robotul lui Torres era specializat în jocul de șah și putea juca doar finalurile rege-turn contra rege. Tabla de șah avea un circuit electric integrat într-o grilă de 8x8. Atunci când regele negru era mutat manual pe tabla, acesta închidea un circuit electric, transmitând astfel poziția sa în algoritmul creat de Torres.

Algoritmul calcula apoi următoarea mutare pe baza poziției actuale și a regulilor șahului, în căutarea unei soluții care să ducă la șah mat. Astfel, robotul putea anticipa mișcările următoare și găsi strategii optime pentru finalurile rege-turn contra rege.

Realizarea acestui robot a reprezentat un pas important în dezvoltarea calculatoarelor și a inteligenței artificiale. A deschis calea pentru cercetările ulterioare și a demonstrat potențialul calculatoarelor de a juca și rezolva probleme complexe.

Confruntarea mileniului a avut loc în anul 1997 și a marcat un moment istoric în domeniul șahului. Aceasta a fost partida disputată între supercalculatorul Deep Blue de la IBM și campionul mondial de șah de la acea vreme, Garry Kasparov.[3]

IBM a investit eforturi considerabile pentru a perfecționa și îmbunătăți performanțele lui Deep Blue între prima întâlnire cu Kasparov în 1996 și partida oficială din 1997. Echipa IBM a lucrat asiduu pentru a actualiza algoritmul și a introduce noi elemente considerate relevante de către operatori în timpul partidelor.

În cele cinci partide disputate între Kasparov și Deep Blue, supercalculatorul a reușit să învingă campionul mondial cu scorul de 3,5 la 2,5. Acest rezultat a marcat prima dată în istorie când un calculator a reușit să depășească abilitățile umane în propriul joc.

Confruntarea mileniului a atras atenția lumii asupra capacităților extraordinare ale inteligenței artificiale și a pus în lumină importanța tehnologiei în progresul jocului de șah.

Acest eveniment a deschis noi orizonturi în cercetarea inteligenței artificiale și a avut un impact semnificativ asupra dezvoltării ulterioare a programelor și roboților de șah.



*Figura 2 Garry Kasparov impotriva lui Deep Blue*

În prezent, există două tipuri principale de roboți jucători de șah. Primul tip constă în roboți industriali de tip manipulator, cărora li se atașează un algoritm de joc specific. Acești roboți sunt de obicei echipați cu brațe articulate și încheieturi flexibile, care le permit să manipuleze piesele de șah și să efectueze mutările pe tabla de joc. Algoritmul de joc este implementat într-un limbaj de programare al roboților, care îi permite să analizeze poziția și să ia decizii de joc în funcție de strategia programată.[4]

Al doilea tip de roboți jucători de șah sunt sistemele integrate magnetic, care funcționează sub tabla de șah și utilizează senzori magnetici de tip Hall. Aceste sisteme sunt compuse dintr-o rețea de senzori plasați strategic sub fiecare pătrat al tablei de șah. Fiecare piesă de șah este echipată cu un magnet corespunzător, iar mișcările pieselor sunt detectate și înregistrate de senzorii magnetici. Informațiile sunt apoi transmise către un sistem de calcul care analizează poziția și efectuează mutările în conformitate cu algoritmul de joc specific.[5]

## 1.2 Motivatie

În contextul actual șahul în format digital a preluat o mare parte din plaja jucătorilor de șah care în trecut se antrenau cu tabla și piesele, însă șahul pe calculator nu are toate elementele care dau acestui joc individualitate. Să joci șah fizic pe tabla de șah îți da o concentrare mai buna asupra jocului, stimulează gândirea de lungă durată și anduranța.

Sistemul mecatronic propus pentru construirea unui robot capabil de a juca șah are ca scop combinarea cunoștințelor interdisciplinare din domeniul proiectării, programării și controlului sistemelor. Acest proiect își propune să ofere o abordare științifică pentru a recrea elementele și avantajele jocului fizic de șah într-un mediu tehnologic.

Construcția acestui robot oferă utilizatorilor posibilitatea de a experimenta jocul de șah într-un mod similar cu cel al unui calculator, dar cu o abordare bazată pe mișcările umane. Această abordare combină aspectele de gândire strategică ale unui calculator cu mișcările naturale ale unei persoane, adăugând o nouă dimensiune și o interacțiune autentică cu jocul.

Dezvoltarea acestui sistem mecatronic reprezintă o oportunitate de explorare și aplicare a principiilor științifice în domeniul roboticii și mecatronicii. Prin integrarea cunoștințelor din domeniile proiectării, programării și controlului sistemelor, se urmărește crearea unui sistem eficient și performant, capabil să gestioneze complexitatea și dinamica jocului de șah.

În final un astfel de proiect care implică roboți jucători de șah poate inspira generația mai tânără. Utilizarea exemplelor interactive și a tehnologiilor avansate într-un joc precum șahul poate captiva atenția și stârni interesul copiilor și tinerilor în domeniul roboticii și automatizării.

## 1.3 Obiective

Pornind de la scopul enunțat anterior putem sparge sarcina mare în mai multe sarcini mai mici grupate în funcție de similarități, pentru a executa acest proiect vor fi definite soluțiile de implementare testare și apoi se vor relua succesiv acești doi pași până se obține un rezultat mulțumitor.

Din punct de vedere al clasificării, sarcinile robotului pot fi împărțite în 2 mari categorii: sarcini hardware și sarcini software, acestea pot fi reprezentate ca în următorul tabel:

*Tabel 1 Sarcini Propuse*

Hardware	Coborare magnet suficient cat pentru a opri atractia magnetica
	Obtinerea unei precizii ridicate
	Asezarea piesei in centrul campului dorit
Software	Impartirea tablei in 64 de patrate
	Preluarea starii unui camp dorit de pe table de sah
	Memorarea pozitiilor din timpul jocului in functie de pozitia initiala
	Crearea unui traseu optim pentru piesa de șah care urmeaza sa fie mutata
	Generarea fisierului cu cod G responsabil pentru comanda motoarelor

Sarcinile hardware pot fi implementate cu ușurință prin utilizarea unei proiectări adecvate, care să țină cont de flambajele interne ale pieselor realizate prin imprimarea 3D în PLA. De asemenea, se va realiza un calcul al dimensiunilor robotului pentru a se asigura compatibilitatea cu dimensiunile tablei de șah.

Sarcinile de tip software vor fi discutate în detaliu în Capitolul 3, unde vor fi prezentați algoritmi implementați pentru abordarea fiecărei probleme în parte. Aici se vor exemplifica

metodele și tehnicile utilizate pentru a rezolva sarcinile software specifice, adaptate la cerințele proiectului.

Prin atenția acordată proiectării hardware și dezvoltării software, se urmărește realizarea unei soluții integrate și eficiente, care să îndeplinească cu succes cerințele și obiectivele propuse în proiectul propus.

#### 1.4 Tehnologii necesare

Pentru a îndeplini sarcinile propuse anterior, ne vom baza pe tehnologii existente pentru a simplifica implementarea soluțiilor găsite în pseudocod. În ceea ce privește proiectarea, vom utiliza software-uri de grafică asistată de calculator, cum ar fi Catia, pentru a crea design-ul pieselor care vor face parte din ansamblul final al robotului.

Pentru controlul robotului, putem opta pentru calcularea directă a geometriei necesare pentru fiecare mutare și situație, dar această abordare ar necesita resurse computaționale și timp considerabile. În schimb, putem abstractiza acest proces prin utilizarea codului G (cod geometric), care primește coordonate carteziane în milimetri și calculează automat tensiunile motoarelor necesare pentru a muta efectorul către destinația dorită.

Pentru procesarea imaginilor, vom utiliza limbajul de programare Python, un limbaj de nivel înalt care ne permite să scriem un număr minim de linii de cod pentru a realiza o logică specifică. Python conține, de asemenea, pachete și librării menite să faciliteze calculul algebric multidimensional, cum ar fi NumPy, Matplotlib, OpenCv, Pandas.

Această abordare combină utilizarea tehnologiilor existente și a instrumentelor software specializate pentru a eficientiza procesul de proiectare, control și procesare a imaginilor în cadrul implementării robotului.

#### 1.5 Structura pe capitole

În continuare, voi prezenta structura acestei lucrări de licență. Capitolul 1 este introducerea proiectului și descrierea strategiilor utilizate pentru realizarea acestuia. Fiecare capitol se conectează cu următorul într-o manieră logică și coerentă.

În Capitolul 2, voi discuta aspectele teoretice studiate în vederea dezvoltării robotului propus. Voi aborda funcționarea motoarelor de șah, teoria jocurilor și modul de funcționare a

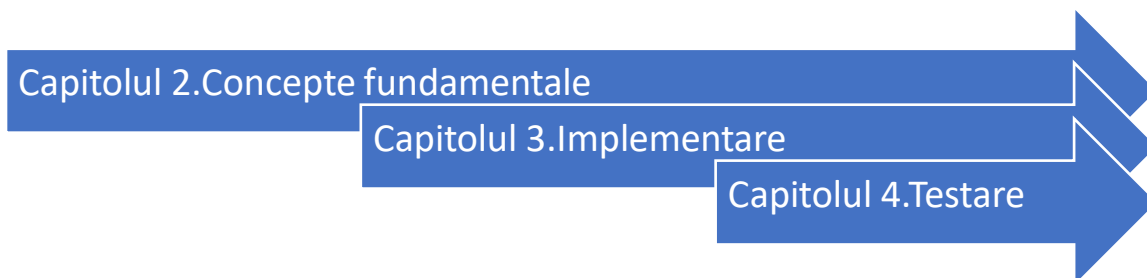


motoarelor populare de șah, cum ar fi Stockfish. De asemenea, voi explora strategiile de extragere a informațiilor și modul de reprezentare al imaginilor.

În Capitolul 3, vor fi prezentate soluțiile constructive și codul implementat, având ca scop rezolvarea sarcinilor propuse. Vor fi abordate aspecte legate de proiectarea modulelor de translatare în cele 3 dimensiuni, proiectarea sistemului electric și implementarea codului și funcțiilor necesare pentru extragerea informațiilor din imagini și generarea răspunsurilor.

Capitolul 4 cuprinde rezultatele experimentale obținute, posibilele direcții viitoare de dezvoltare și concluziile proiectului. Aici vor fi prezentate teste referitoare la mișcarea robotului pe sub tabla de șah, informațiile extrase din imagini și generarea codului care conține mutările răspuns.

Prin această structură, se urmărește o prezentare coerentă și detaliată a tuturor aspectelor relevante ale proiectului, începând de la fundamentarea teoretică până la implementarea și rezultatele obținute.



*Figura 3 Structura pe capitole*

## 2 Concepte fundamentale

---

Jocul de șah

Motoare de șah

Teoria jocurilor

Stockfish

Interpretarea scenei

---

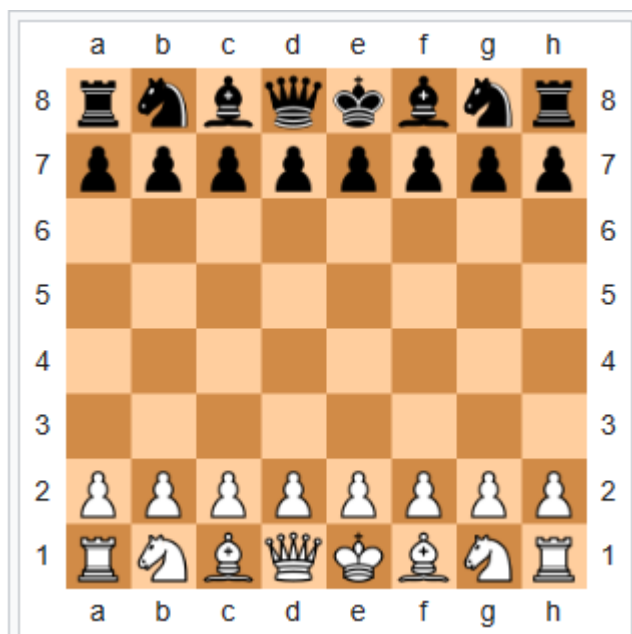
„Șahul nu este un simplu joc. Șahul este o formă bine definită de computație. Poate că nu puteți găsi răspunsurile, dar în teorie, trebuie să existe o soluție, o procedură corectă pentru orice poziție”

- John von Neumann.

## 2.1 Jocul de șah

Șahul este un joc de strategie abstractă jucat între doi jucători, Alb și Negru, pe o tablă de șah cu 64 de pătrățele aranjate într-un grid de opt pe opt. La început, fiecare jucător controlează șaisprezece piese: un rege, o regină, două turnuri, doi nebuni, doi cai și opt pioni. Jocul este câștigat prin punerea regelui advers în poziție de șah-mat, adică amenințându-l cu capturarea inevitabilă. Există mai multe modalități în care un joc se poate încheia egal[6].

La începutul partidei de șah piesele sunt mereu aranjate în aceeași poziție numită poziție de start de unde jucătorul pieselor albe va începe partida având dreptul de a executa prima mutare[6].



*Figura 4 Poziție initială a pieselor de șah*

Scopul lucrării este de a realiza un robot capabil de a muta piesele corect în conformitate cu regulile jocului și cu o precizie ridicată pe tabla de șah, însă pentru partea inteligentă a robotului există mai multe soluții care pot fi luate în considerare printre care: motoarele de șah, teoria jocurilor sau modelele matematice.

## 2.2 Motoare de șah

Un motor de șah este un program de calculator sau o aplicație specializată care este concepută pentru a juca șah. Este denumit și "program de șah" sau "program de șah pe calculator". Acesta utilizează algoritmi euristici pentru a analiza pozițiile de șah, a evalua mutările și a lua decizii optime în timpul jocului.

Un motor de șah funcționează prin utilizarea unei combinații de tehnici și strategii de calcul pentru a genera și explora un arbore de joc extins. Aceasta înseamnă că motorul examinează diferite linii de joc posibile și calculează consecințele acestora pe baza unor reguli specifice și a unei evaluări a poziției.

Motorul de șah utilizează diferite componente și tehnici pentru a-și îndeplini funcțiile. Printre acestea se numără:

**Evaluarea poziției:** Motorul evaluează poziția curentă a pieselor pe baza unor factori precum valoarea materială, structura pionilor, siguranța regelui, activitatea pieselor și controlul centrului tablei. Acest lucru se face cu ajutorul unei funcții de evaluare care atribuie un scor numeric poziției și ajută la luarea deciziilor[7].

**Algoritmi de căutare:** Motorul utilizează algoritmi avansați de căutare pentru a explora diferitele mutări posibile și a găsi cea mai bună mutare. Unul dintre algoritmii de căutare utilizați este algoritmul minimax cu tăiere alpha-beta[8], care optimizează căutarea prin eliminarea ramurilor arborelui care nu sunt relevante pentru luarea deciziilor optime.

**Baze de date și deschideri:** Motorul de șah utilizează adesea baze de date cu poziții și mutări deschise, care sunt combinații predefinite de mutări în faza de început a jocului.

Aceasta ajută motorul să ia decizii rapide și să evite recalcularea mutărilor deja cunoscute. **Optimizări și tehnici avansate:** Motorul de șah poate folosi diverse optimizări și tehnici pentru a îmbunătăți performanța sa. Acestea includ utilizarea tabelurilor de transpoziție (caching-ul pozițiilor analizate anterior), utilizarea tabelurilor de final de joc (pentru poziții de final perfecte precalculate), folosirea procesării paralele (utilizarea mai multor nuclee de procesor pentru a accelera calculele) și altele.

Motorul de șah este utilizat într-o varietate de contexte, cum ar fi competiții de șah online, analiză și studiu individual al partidelor de șah și dezvoltarea de aplicații și interfețe de șah pentru jucători. Prin puterea lor de calcul și cunoștințele

## 2.3 Teoria Jocurilor

Teoria jocurilor este reprezentată de studiul modelelor matematice și a interacțiunilor strategice care se produc între agenți raționali. Din punct de vedere al teoriei jocurilor, șahul este un joc de tipul suma-zero, cu informație perfectă de tip combinatorial. [9]

Un joc de tip suma zero înseamnă că deciziile executate de jucători nu pot nici să crească dar nici să scadă cantitatea de resurse disponibile. În jocurile de tip suma zero beneficiile sunt distribuite în mod egal către toți participanții la joc, orice combinație de strategii duce întotdeauna la o suma zero. Șahul este un joc de tip suma zero deoarece orice avantaj obținut provine în urma unui dezavantaj generat oponentului.

*Tabel 2 Un joc de tip suma 0*

	A	B
A	-1,1	3,-3
B	0,0	-2,2

Șahul este de asemenea un joc secvențial cu informații perfecte. Un joc cu informații perfecte înseamnă că toți jucătorii, la fiecare mutare în joc, cunosc istoricul anterior al jocului și mutările făcute anterior de toți ceilalți jucători.

Jocurile în care dificultatea găsirii unei strategii optime provine din multiplicitatea mutărilor posibile se numesc jocuri combinatorii. Jocurile cu informații perfecte au fost studiate în teoria combinatorială a jocurilor, care a dezvoltat reprezentări noi, cum ar fi numerele ireale, precum și metode de demonstrație combinatorii, algebrice (și uneori non-constructive) pentru rezolvarea anumitor tipuri de jocuri, inclusiv jocuri "cu bucle" care pot rezulta în secvențe infinite de lungi de mutări.

## 2.4 StockFish

Stockfish este cel mai popular și puternic motor de șah de la ora actuală, poate fi numit state of the art în domeniul șahului. Acesta utilizează algoritmul de căutare alpha-beta

pruning [10]. Alpha-beta pruning îmbunătățește căutarea minimax [minMax] prin evitarea variantelor care nu vor fi atinse în jocul optim, deoarece unul dintre jucători va redirectiona jocul.

Deoarece este adesea nefezabil din punct de vedere computațional să se caute până la sfârșitul jocului, căutarea este întreruptă într-un stadiu timpuriu atunci când ajunge la o anumită adâncime. Adâncimea căutării este măsurată în play-uri, unde un play reprezintă o tură efectuată de un jucător. O adâncime de căutare de  $D$  indică faptul că distanța între nodul rădăcină și nodurile frunză ale arborelui de căutare este de  $D$  plays. Stockfish își crește treptat adâncimea arborelui său de căutare într-un proces cunoscut sub denumirea de adâncire iterativă [11]. Cu toate acestea, atunci când motoarele de șah raportează o adâncime nominală de  $D$ , aceasta nu înseamnă că căutarea a luat în considerare toate variațiile posibile ale  $D$  mutări. Aceasta se datorează euristicilor care determină motorul să caute variații promițătoare la o adâncime mai mare decât nominal și variații mai puțin promițătoare la o adâncime mai mică decât nominal.

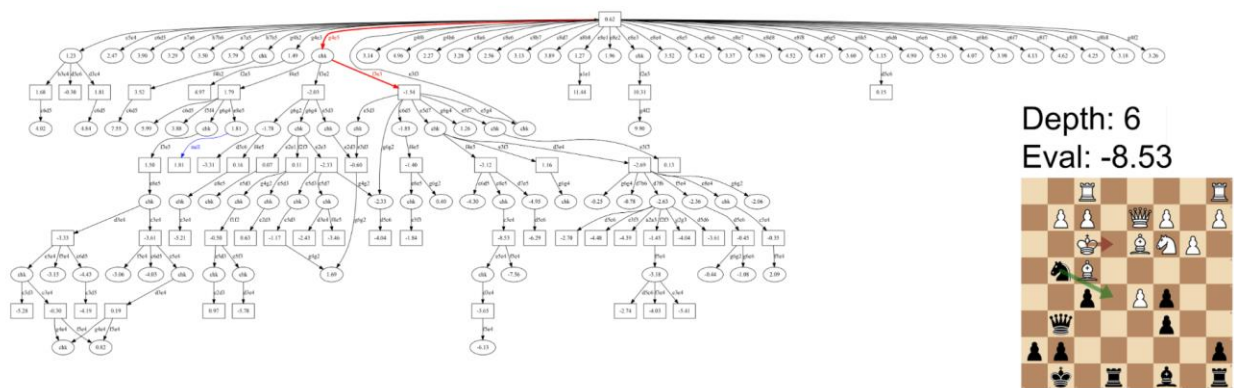


Figura 5 Arbore de cautare

Motorul aplică două clase principale de euristici pentru a reduce spațiul de căutare: tăierea și reducerea înaintată [12]). Tehnicile de tăiere înaintată elimină subgrafurile arborelui de joc care nu sunt susceptibile să fie prezente în jocul optim. De exemplu, dacă evaluarea unei poziții este semnificativ mai proastă decât valoarea garantată de cea mai bună alternativă a unui jucător, copii ale poziției sunt tăiate prematur. Aceasta se numește tăiere inutilă [13]. Este posibil ca motorul să taie în mod greșit o linie de joc. Aceasta va fi corectată

odată ce adâncimea motorului depășește o limită specifică tehnicii, după care tehnica nu mai este aplicată.

Tehnicile de reducere caută anumite subgrafuri ale arborelui de joc la adâncimi mai mici, în loc să le omită complet din căutare. Un exemplu clasic este reducerea mutărilor târzii [14] care presupune că motorul verifică mai întâi mutări mai bune. Mutările verificate mai târziu sunt căutate la adâncimi mai mici decât cele nominale.

Dând timp infinit, motorul va converge către linia optimă de joc. Limitele de adâncime previn din ignorarea liniilor prin tăiere, iar reducerile devin nesemnificative la adâncimea infinită.

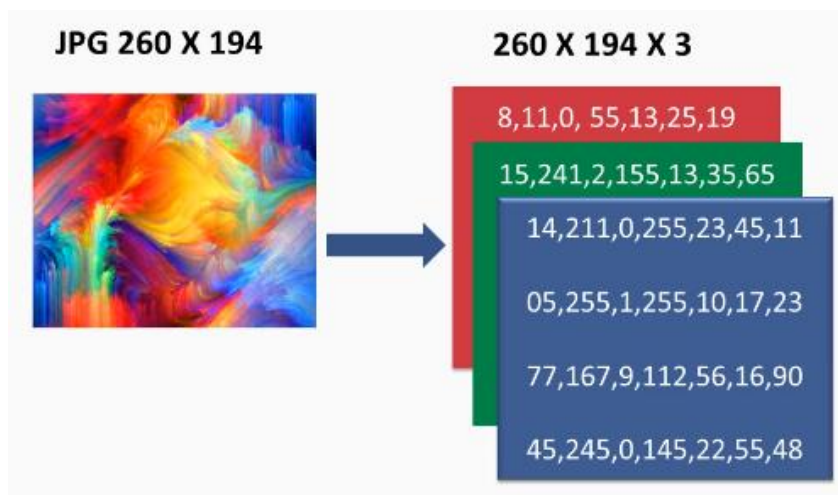
## 2.5 Interpretarea scenei

Pentru un calculator, o imagine coloră este o matrice caracterizată de dimensiuni și de numărul canalelor de culoare ale acesteia, culoarea în format digital poate fi reprezentată sub formă de nuanțe de gri de diferite precizii sau color în diferite formate RGB, BGR, HSV etc.

În cazul matricilor colore avem de a face cu 3 matrici diferite, una pentru fiecare culoare primară aditivă din spectrul de frecvență.

Din punct de vedere matematic, o imagine coloră poate fi de asemenea reprezentată ca o funcție de tipul:

$$F(x,y):R^2 \rightarrow R^3 \quad (1)$$



*Figura 6 Reprezentare a imagini color in calculator[6]*

## 2.6 Codul G

Codul G, prescurtare de la "Geometric Code", este un limbaj de programare utilizat pentru controlul mașinilor cu comandă numerică pe calculator (CNC), precum imprimantele 3D, mașinile de frezat și strungurile. El constă într-o serie de comenzi și instrucțiuni care îi indică mașinii cum să se miște, să se poziționeze și să opereze diverse componente.

Codul G servește ca un limbaj standardizat care facilitează comunicarea între un calculator și o mașină CNC. El utilizează o combinație de litere, numere și caractere speciale pentru a reprezenta diferite acțiuni și parametri. Fiecare comandă în cod G corespunde unei operații specifice, cum ar fi deplasarea mașinii de-a lungul unei axe, setarea vitezelor de deplasare, activarea/dezactivarea uneltelor și controlul răcirii sau a vitezei de rotație a axului principal.[16]

Structura codului G urmează în mod obișnuit un format secvențial, în care fiecare comandă este listată pe o linie separată. Fiecare linie începe cu litera "G" sau "M", indicând dacă este o comandă de mișcare, respectiv o comandă diversă. După literă, se utilizează valori numerice pentru a specifica parametri precum coordonatele, vitezele de avans, numerele de unelte și decalaje.

Codul G permite controlul precis al mașinilor CNC, permițând operații complexe. Acesta poate fi generat manual sau prin intermediul software-urilor de proiectare asistată de



calculator (CAD) sau fabricație asistată de calculator (CAM). Cu ajutorul G-code, utilizatorii pot programa traiectorii de unelte complexe, pot crea forme geometrice, pot executa sarcini repetitive și pot produce design-uri complexe cu un grad ridicat de precizie și repetabilitate.

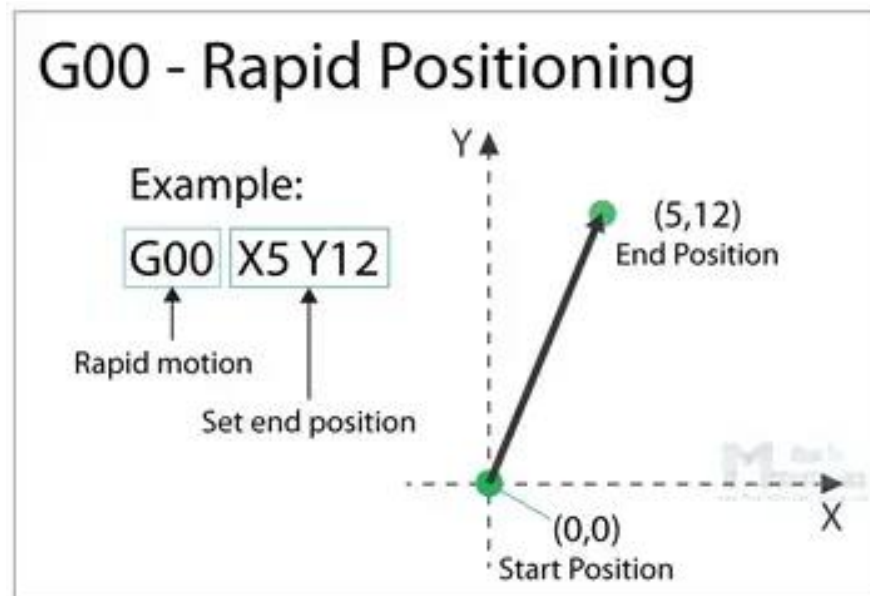


Figura 7 Mișcare liniară în cod G

## 3 Conținut

---

Proiectarea sistemului mecatronic

Proiectarea sistemului electric

Procesarea Imagini

Generarea de cod G

---

### 3.1 Proiectarea Sistemului Mecatronic

Sistemul mecatronic proiectat dispune de un spațiu de lucru generos, având forma unui dreptunghi cu dimensiunile de 60 cm pe 50 cm. Acest spațiu este ideal pentru a găzdui și manipula piesele de șah în cadrul jocului. Carcasa robotului, care protejează și susține întregul sistem, este compusă dintr-o placă suport cu dimensiunile de 80 cm pe 60 cm. Această placă suport oferă stabilitate și rezistență în timpul funcționării robotului.

Pentru a susține tabla de șah în poziția corespunzătoare, sunt atasate plăci suport în două puncte, asigurând astfel o fixare fermă și uniformă pe întreaga lungime a tablei. Aceste suporturi contribuie la stabilitatea și poziționarea precisă a tablei de șah, oferind un mediu optim pentru desfășurarea jocului.

Robotul propriu-zis este alcătuit din trei module de translație, dintre care două se deplasează în planul XoY și realizează mișcarea pieselor de șah pe suprafața tablei. Aceste module de translație permit robotului să se deplaseze într-un mod precis și controlat, asigurând mutarea pieselor în casuțele dorite.

Al treilea modul de translație este destinat mișcării pe axa Z și este plasat la capătul robotului. Acest modul de translație permite ridicarea și coborârea unui magnet permanent, care este atașat la capătul opus al acestui modul. Prin intermediul acestui magnet permanent, se realizează manipularea pieselor de pe partea opusă a tablei de șah, asigurând astfel posibilitatea de a muta și de a poziționa piesele în mod eficient și precis.

În ansamblu, sistemul mecatronic dezvoltat prezintă o configurație bine gândită și eficientă, oferind un spațiu de lucru optim, o carcasa solidă și module de translație avansate, care permit manipularea pieselor de șah într-un mod controlat și precis. Această abordare tehnologică aduce o nouă dimensiune în jocul de șah, oferind o experiență captivantă și interactivă pentru utilizatori.

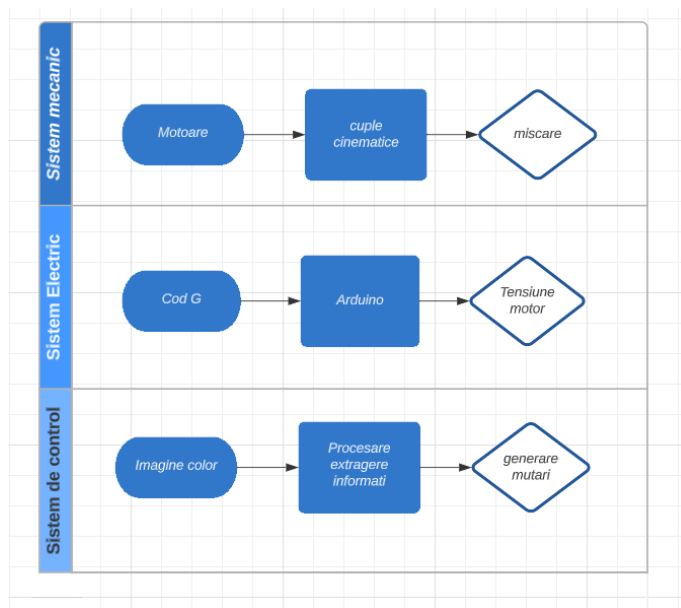


Figura 8 Diagrama bloc

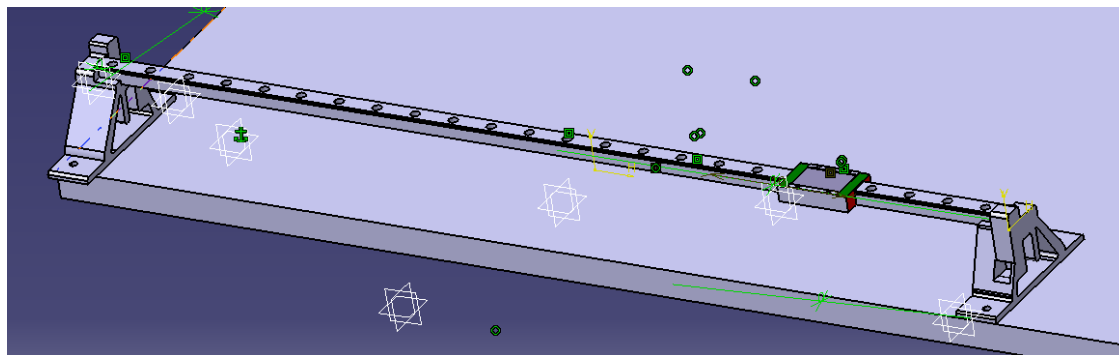
### 3.1.1 Modul de translație axa X

Modul de translație pe axa X este compus dintr-un ghidaj liniar cu patină de tipul MGN12, care asigură o mișcare lină și precisă. Acest ghidaj liniar este fixat în doi suportați imprimati 3D, unul dintre ei găzduind motorul pas cu pas, iar celălalt fiind poziționat la capătul cursei și având o extensie verticală pentru limitatorul de capăt de cursă.

Patina ghidajului liniar, de tip MGN12c, este un rulment liniar cu bile de înaltă calitate, care garantează o funcționare fără întreruperi și o stabilitate excelentă în timpul mișcărilor. Pentru a atașa modulul de translație de pe axa Y la această patină, se utilizează o piesă specială, imprimată 3D, și șuruburi M3 pentru a asigura o fixare solidă și precisă.

Motorul de acționare al modulului de translație este un motor pas cu pas NEMA 17, recunoscut pentru performanțele sale fiabile și cuplul său optim. Cu un cuplu de 0,42 N/m, acest motor este suficient de puternic pentru a pune în mișcare cele două module atașate, chiar și cu o greutate ridicată. Transmiterea mișcării către rulmentul liniar se realizează printr-o curea de cauciuc dințată, care este fixată pe un scripete cu dinți amplasat la capătul motorului, atașat de un alt scripete la celălalt capăt al piesei și încastrată în cuplajul de prindere dintre rulment și axa Y, asigurând o transferare eficientă și precisă a mișcării.

Această configurație complexă și bine gândită a modulului de translație pe axa X permite robotului să efectueze mișcări precise și controlate, asigurând stabilitatea și performanța necesare într-o gamă variată de aplicații robotice.



*Figura 9 Modul de translație I*

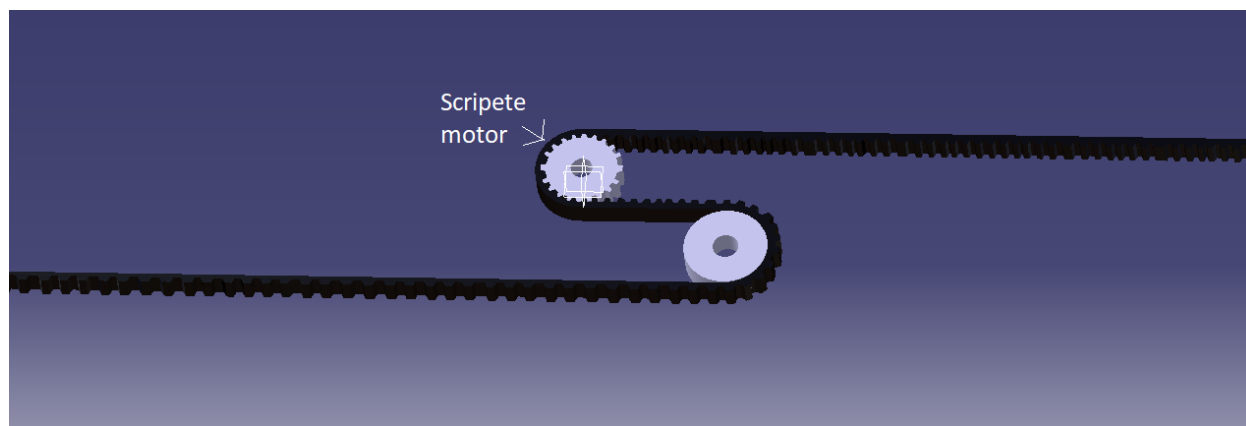
### 3.1.2 Modul de translație de pe axa Y

Modulul de translație de pe axa Y este alcătuit din mai multe componente esențiale pentru funcționarea sa eficientă. În primul rând, avem o piesă imprimată 3D special concepută pentru a realiza cuplarea între modulul de translație pe axa X și modulul de

translație pe axa Y. Această piesă asigură o conexiune sigură și precisă între cele două module, garantând stabilitatea întregului sistem.

Pentru a susține mișcarea pe axa Y, sunt utilizate două tije de oțel care sunt fixate în piesa imprimată 3D de la un capăt. Aici este amplasat un servomotor, care acționează tijele de oțel și modulul de translație pe axa Z, care se găsește la celălalt capăt al modulului de translație pe axa Y. Această configurare permite realizarea mișcărilor precise și controlate pe axa Y.

Piesa de cuplaj dintre modulul de translație de pe axa X și axa Y are un rol crucial în întregul ansamblu. Pe lângă conectarea celor două module, aceasta îndeplinește multiple funcții esențiale. Piesa susține motorul pas cu pas asociat modulului 2 și are două canale în partea de jos, în care sunt fixate capetele curelei dintate care transmit mișcarea către modulul 1. De asemenea, piesa dispune de găuri de fixare pentru cei doi scripeți prin care trece cureaua de transmisie a mișcării modulului 2. Aceasta are, de asemenea, două locașuri în care sunt inserați rulmenții liniari pentru susținerea tijelor de oțel și include un suport pentru limitatorul de cursă, asigurând un control precis al mișcării.



*Figura 10 Mecanism transmitere miscare modul II*

A doua piesă printată 3D din componenta modulului II se găsește la capatul posterior al modulului și are rolul de a fixa rigid un capăt al curelei cu dinți care transmite mișcarea motorului la modul, acesta este antrenat de către curea și împinge întregul modul în față sau în spate în funcție de mișcarea motorului, de asemenea prezintă și un locaș pentru montarea servomotorului aferent modulului III care este plasat la capatul opus.

### 3.1.3 Modul de translație axa Z

Modulul III este special conceput pentru a realiza translația pe axa Z și îndeplinește o funcție importantă în sistemul robotic destinat jocului de șah. Scopul acestui modul este de a permite cuplarea și decuplarea pieselor de șah sub tabla de joc. Acest lucru este realizat prin translatarea unui magnet permanent pe axa Z, astfel încât să se apropie suficient de mult de un magnet plasat pe partea opusă a tablei de șah, generând astfel o forță magnetică ce permite mișcarea pieselor.

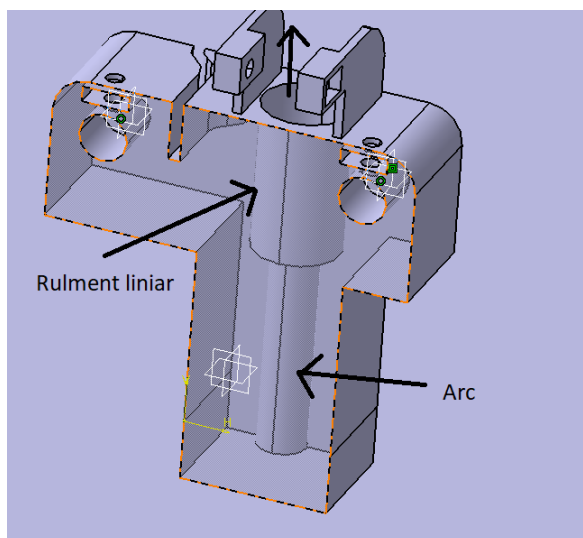


Figura 11 Privere în interior a modului de translație

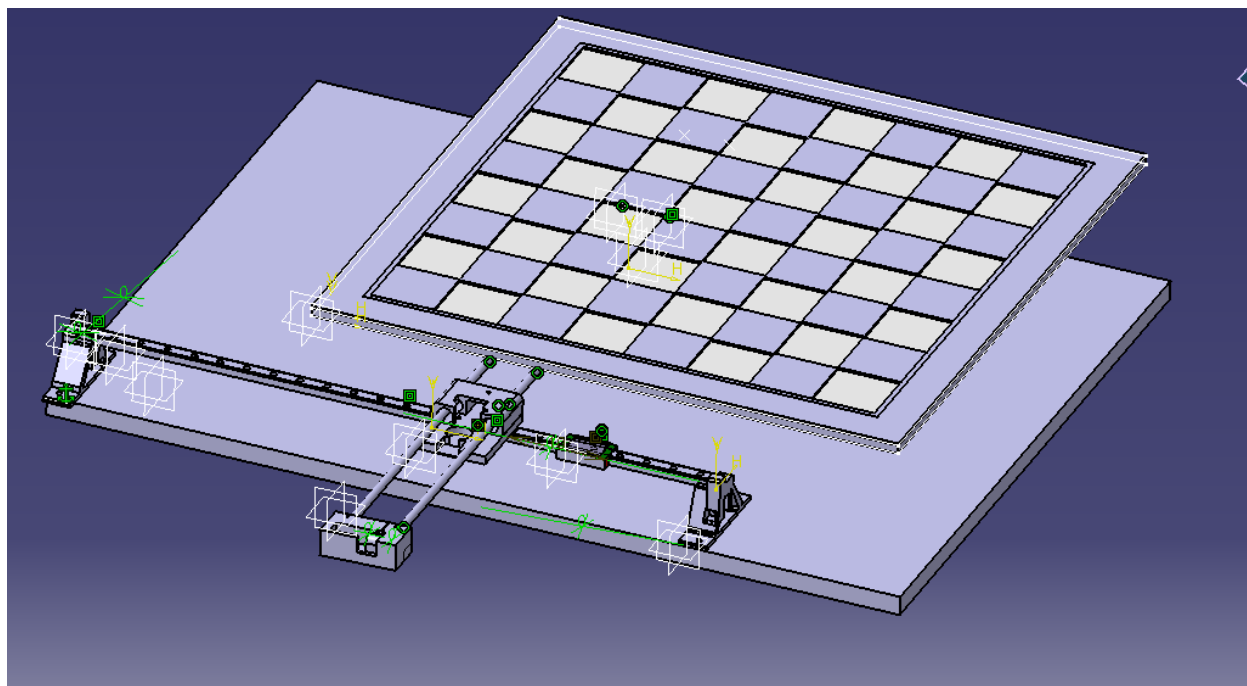
Modulul III este alcătuit din mai multe componente esențiale. În primul rând, avem o piesă imprimată 3D care joacă un rol central în acest modul. Această piesă este compusă din două tronsoane distincte. În primul tronson, cu un diametru de 10 mm, se introduce un arc de oțel. Acest arc are rolul de a oferi o susținere elastică în poziția normală a modului, ridicând tija de oțel și aducând magnetul în contact cu tabla de șah și piesele de pe aceasta. În această poziție, piesele de șah pot fi mutate cu ajutorul robotului și pot fi transportate la orice punct de pe tabla de joc.

În al doilea tronson al piesei imprimate 3D, este plasat un rulment liniar. Acest rulment permite tijei de oțel, cu o lungime de 8 cm, să se deplaseze în direcția axei Z. Astfel,

mișcarea tijei de oțel este controlată și permite ajustarea poziției magnetului în funcție de necesități.

Pentru a realiza mișcarea de translație pe axa Z, este utilizat un servomotor care este atașat la capătul opus al modulului de translație II. Acest servomotor acționează o curea dințată care este lipită de tija de oțel și trecută prin intermediul unui scripete fixat pe piesa imprimată 3D. Prin acționarea servomotorului, direcția de orientare a curelei dințate este schimbată de la orientarea pe axa Y la orientarea pe axa Z, permițând astfel mișcarea magnetului și realizarea translației pe axa Z.

Configurația complexă și ingenioasă a modulului III permite manipularea și poziționarea precisă a pieselor de șah, asigurând o experiență de joc fluidă și precisă în cadrul sistemului robotic.



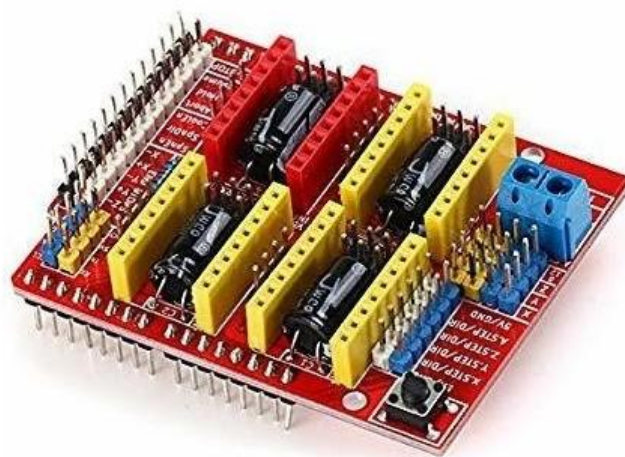
*Figura 12 Robot Asamblat*

### 3.2 Proiectarea Sistemului Electric

Sistemul electric este responsabil pentru transmiterea comenzilor de la calculator la motoarele robotului dar și pentru primirea de informații despre poziția modulelor de translație ale robotului.

Sistemul electronic al robotului este compus dintr-un motor pas cu pas, un scut motor de tip CNC, Arduino Uno și limitatoare de cursă, aceasta construcție reprezintă o configurație comună utilizată în diverse aplicații de robotică.

Motorul pas cu pas este un tip special de motor electric utilizat pentru controlul precis al mișcărilor. Acesta este alcătuit din bobine și poli magnetici care permit poziționarea precisă în etape discrete. Prin aplicarea unor secvențe de curent la bobine, motorul pas cu pas poate fi controlat pentru a efectua rotații precise și repetitive. Acest tip de motor este folosit frecvent în roboți și echipamente CNC, unde precizia mișcărilor este esențială.



*Figura 13 Scut Motor de tip CNC*

Scutul-ul CNC este o placă de expansiune care se conectează la Arduino Uno și oferă funcționalități specifice pentru controlul motoarelor și interfeței cu echipamentele CNC (Computer Numeric Control). Acesta oferă conexiuni pentru motoarele pas cu pas și limitatoarele de cursă, precum și alte funcții importante pentru controlul mișcărilor și comanda robotului.

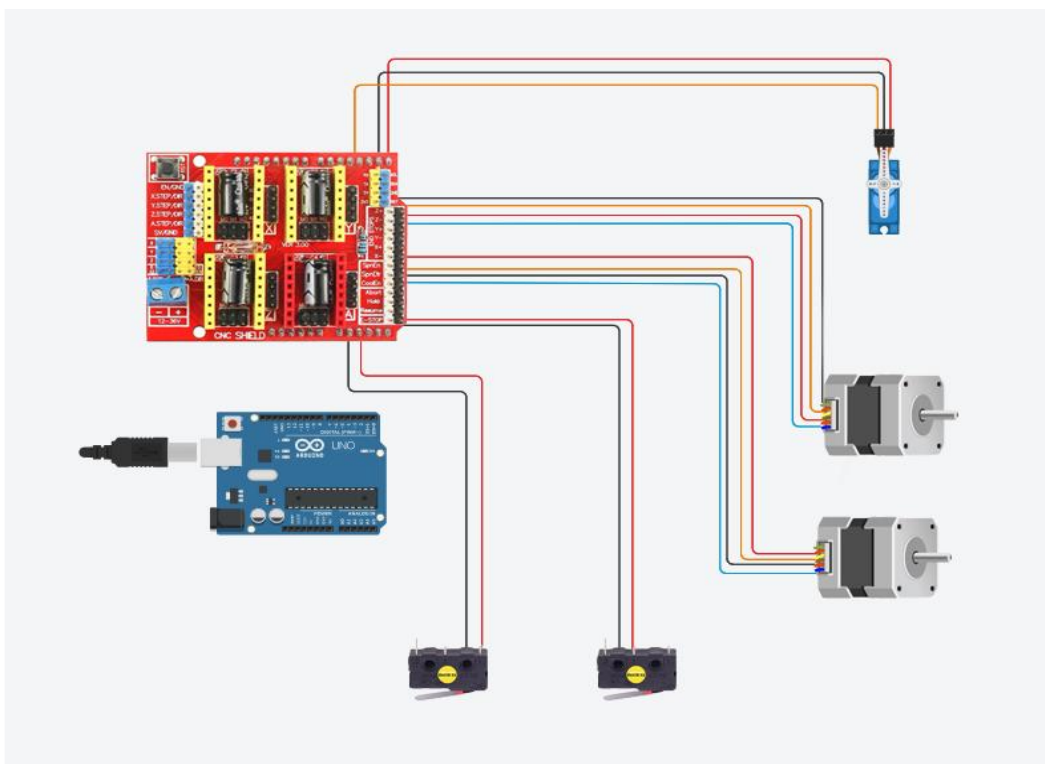
Arduino Uno este o platformă de dezvoltare open-source bazată pe microcontrollerul cu același nume. Este utilizată într-o gamă largă de aplicații, inclusiv robotică. Arduino Uno oferă un mediu de programare ușor de utilizat și posibilități de interfațare cu diverse componente electronice și senzori. În cadrul sistemului robotic, Arduino Uno preia comenzile



și funcționează precum creierul sistemului, controlând motorul pas cu pas și primind informații de la limitatoarele de cursă.

Limitatoarele de cursă, numite și switchuri, sunt dispozitive electronice sau mecanice care detectează poziția maximă sau minimă a unei componente sau a unui mecanism într-o anumită direcție de mișcare. Acestea sunt utilizate pentru a asigura controlul precis al mișcărilor și pentru a preveni depășirea limitelor fizice ale robotului. Atunci când limitatorul de cursă este activat, acesta trimite semnal către Arduino Uno, care va opri mișcarea motorului pas cu pas în consecință.

Prin combinarea motorului pas cu pas, a scutului motor, a placutei Arduino Uno și limitatoarelor de cursă, se poate realiza un sistem electronic eficient pentru controlul precis al mișcărilor în aplicații robotice. Această configurație permite programarea și controlul robotului, asigurând o navigare precisă și evitarea depășirii limitelor fizice. Este important să se dezvolte un cod de programare adecvat pentru a coordona funcționarea tuturor componentelor și pentru a realiza sarcinile specifice ale robotului în aplicația dorită.



*Figura 14 Schemă circuit electric*

### 3.3 Detectia miscarii pieselor de șah folosind imagini

Imaginile prelevate de camera web sunt incarcate in memorie cu ajutorul modulului openCv dupa care are loc o calibrare manuala a tablei de sah. In scena se transmit ca si parametri pentru algoritmul de detectie a casutelor cordonatele din planul XoY a celor 4 colturi ale tablei de sah. Dupa primirea datelor se realizeaza desenarea unui contur virtual incepand cu colțul din dreapta sus care este notat ca si punctul A si pana la colțul din dreapta jos notat D se obtine noul spatiu de lucru din patrulaterul ABCD.

In functie de poziția camerei vor exista niste deformații ale imaginii. Punctele mai apropiate de camera vor fi percepute ca fiind mai mari in timp ce cele mai departate vor fi mai mici.

#### 3.3.1 Functia de preluare a starii

Acum ca avem noul spatiu de lucru putem incepe trasarea chenarului in care vom construi virtual o noua tabla de șah din care vom incerca sa extragem cat mai multă informație posibilă. Obiectivul principal este de a face diferența dintre un câmp ocupat de o piesa si un câmp in care nu se afla nimic, pentru a realiza acesta sarcină întâi avem nevoie sa ne definim limitele folosind cordonatele primite in urma calibrarii, vom impartii cele doua segmente AB si AC in 8 segmente mai mici, echidistante din care mai apoi vom trasa linii drepte in plan perpendicular fata de segmentul original pentru a realiza grila de 8 pe 8 de care este nevoie.

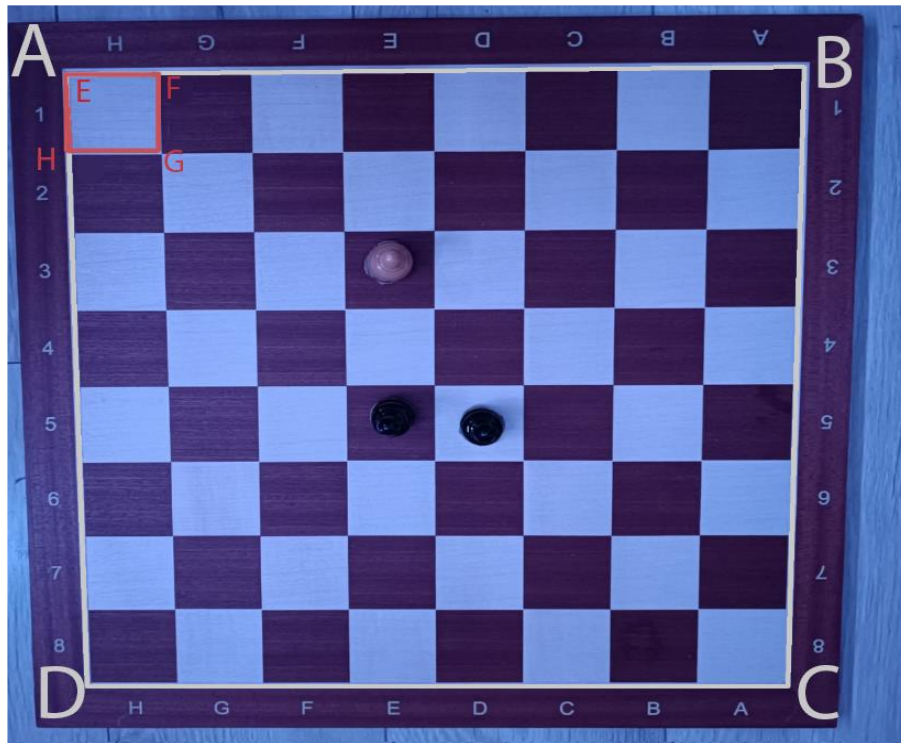


Figure 15 Împartirea tablei de șah in câmpuri

În pasul 2 este necesar calculul mediei tuturor pixelilor din toate câmpurile obținute. Pentru a implementa acest lucru va fi nevoie de un vector de tipul tuple care conține toate cele 4 puncte ale fiecărui pătrat in patru puncte care vor fi denumite mai departe cu E, F, G respectiv H.

$$E=(AB[i][0],AC[j][1]) \quad (2)$$

$$F=(AB[1+i][0],AC[j][1]) \quad (3)$$

$$G=(AB[i][0],AC[1+j][1]) \quad (4)$$

$$H=(AB[1+i][0],AC[1+j][1]) \quad (5)$$

Aplicand iterativ (2),(3),(4) si (5) obtinem vectorul de coordonate ale campurilor tablei de sah virtuale.

Urmatoarea etapa in procesul este calculul mediei aritmetice a tuturor valorilor pixelilor prezenti in fiecare câmp, medie aritmetica care trebuie calculată in 3 variabile diferite,

cate una aferentă fiecarui canal de culoare, urmând ca mai apoi cele 3 medii sa fie unite intr-un tuplu ce va contine media de culoare a câmpului actual.

### 3.3.2 Functia de generare a codului G

Codul G ii spune robotului in ce punct se poate duce, scopul generatorului de cod g este de a transforma informatia digitala in coordonate milimetrice pentru efector, pentru a realiza aceasta sarcina trebuie sa transforme o mutare din scriere algebrica (e2-e4) in secvente de linii si pauze (G01 x2 y5), mutarile permise se vor afisa intr-un fisier numit mutari.txt, fisier ce va trece printr-o functie de transformare in valoare numerica prin următoarea formulă.

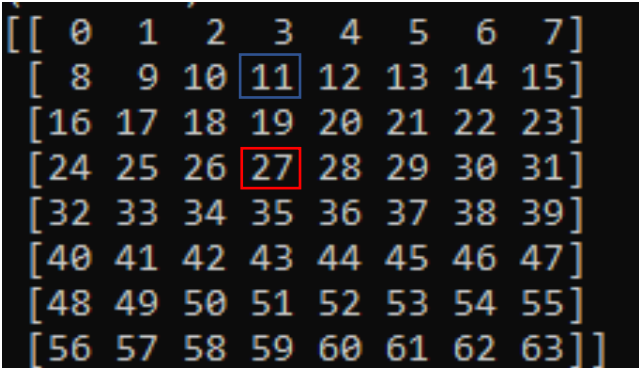
$$Mn = Y + (X - 1) * 8 \quad (6)$$

Unde Mn reprezintă mutarea in format numeric

Y reprezintă coordonata pe axa Y, respectiv coloană pe care se afla piesa.

X reprezintă coordonata pe axa X, respectiv linia pe care se afla piesa.

Dupa convertirea mutarilor in format numeric este necesară cunoasterea diferentei dintre câmpul de plecare al piesei si câmpul de sosire al piesei.



[	0	1	2	3	4	5	6	7]
[	8	9	10	11	12	13	14	15]
[	16	17	18	19	20	21	22	23]
[	24	25	26	27	28	29	30	31]
[	32	33	34	35	36	37	38	39]
[	40	41	42	43	44	45	46	47]
[	48	49	50	51	52	53	54	55]
[	56	57	58	59	60	61	62	63]]

*Figura 16 Tabla de șah reprezentată in memorie. Evidențiat cu chenarul albastru poziția de start a piesei, respectiv de sosire cu chenarul rosu*

Mutările sunt permise in formatul unui tuplu (câmp plecare, câmp sosire). Dupa ce sunt transformate in valori numerice se scade campul de sosire din campul de plecare, iar diferenta este trecută prin urmatoarele formule pentru a obtine numarul de pasi către dreapta, respectiv numarul de pasii ascendenti pentru a ajunge la câmpul de sosire.

$$PD = \text{diff} / 8 \quad (7)$$

$$PS = \text{diff} \% 8$$

(8)

În final se adaugă după fiecare câmp de plecare comanda de cuplare a piesei (m3 s90) care comanda servomotorul aferent modului de translație, să elibereze tijă, care iese din tensiunea curelei este ridicată ascendent de către arc către tabla pentru a cupla piesa de șah. Respectiv comanda (G4 p1) pentru a aștepta o secundă pentru o mișcare mai fluidă și umană a robotului și în final inversul comenzii (m3 s90) care este comanda (m5)

Pentru realizarea mutării calului ne putem defini o nouă matrice de dimensiune 17x17 unde liniile de delimitare dintre coloanele și liniile standard devin cărări posibile de traversare. Din dorința de simplificare a codului se va asuma că întotdeauna traseul calului va fi ocupat de o altă piesă așa că pentru a executa o mutare de cal (b8,c6) nu se va aplica

$(-1 * PD, 2 * PS)$  ci mutările pe axa X vor fi înjumătățite și executate separat odată la plecare și odată la sosire obținând  $(-0.5 * PS, 2 * PD, -0.5 * PS)$

De asemenea, această strategie poate fi folosită și la executarea rocadei mici sau mari, a doua și ultima mutare din șah care necesită ca o piesă să sară peste o altă.

A1	A1.5	A2	A2.5	A3	A3.5	A4	A4.5	A5	A5.5	A6	A6.5	A7	A7.5	A8	
AB1		AB2		AB3		AB4		AB5		AB6		AB7		AB8	
B1	B1.5	B2	B2.5	B3	B3.5	B4	B4.5	B5	B5.5	B6	B6.5	B7	B7.5	B8	
BC1		BC2		BC3		BC4		BC5		BC6		BC7		BC8	
C1	C1.5	C2	C2.5	C3	C3.5	C4	C4.5	C5	C5.5	C6	C6.5	C7	C7.5	C8	
CD1		CD2		CD3		CD4		CD5		CD6		CD7		CD8	
D1	D1.5	D2	D2.5	D3	D3.5	D4	D4.5	D5	D5.5	D6	D6.5	D7	D7.5	D8	
DE1		DE2		DE3		DE4		DE5		DE6		DE7		DE8	
E1	E1.5	E2	E2.5	E3	E3.5	E4	E4.5	E5	E5.5	E6	E6.5	E7	E7.5	E8	
EF1		EF2		EF3		EF4		EF5		EF6		EF7		EF8	
F1	F1.5	F2	F2.5	F3	F3.5	F4	F4.5	F5	F5.5	F6	F6.5	F7	F7.5	F8	
FG1		FG2		FG3		FG4		FG5		FG6		FG7		FG8	
G1	G1.5	G2	G2.5	G3	G3.5	G4	G4.5	G5	G5.5	G6	G6.5	G7	G7.5	G8	
GH1		GH2		GH3		GH4		GH5		GH6		GH7		GH8	
H1	H1.5	H2	H2.5	H3	H3.5	H4	H4.5	H5	H5.5	H6	H6.5	H7	H7.5	H8	

Figura 17 Matrice cu mutări posibile cal

## 4 Concluzi

---

Teste si rezultate obtinute

Obiectivele si functiile obtinute

Utilitatea temei si directii viitoare de dezvoltare

---

### 4.1 Teste si rezultate obtinute

#### 4.1.1 Preluarea stării

In cazul unu avem un exemplu de scena preluata in lumina predominant albastra, împărțirea tablei intr-un chenar virtual nu este realizată 1:1 cu conturul tablei, iar in scena sunt prezente numai 3 piese doi pioni negri si unul alb. Pentru cazul unu in care căutăm câmpurile albe goale obținem 6 fals pozitive si 57 de pozitive adevărate, ceea ce inseamna o precizie de 90.48%, în cazul in care căutam câmpurile negre goale aveam un raport de 60 la 3, adică 95% precizie, o observatie interesanta este că prezenta pionului negru in câmpul alb D5 a fost suficientă pentru a intoarce media de culoare a câmpului si al trece de testul de diferență pentru a fii raportat ca si câmp negru

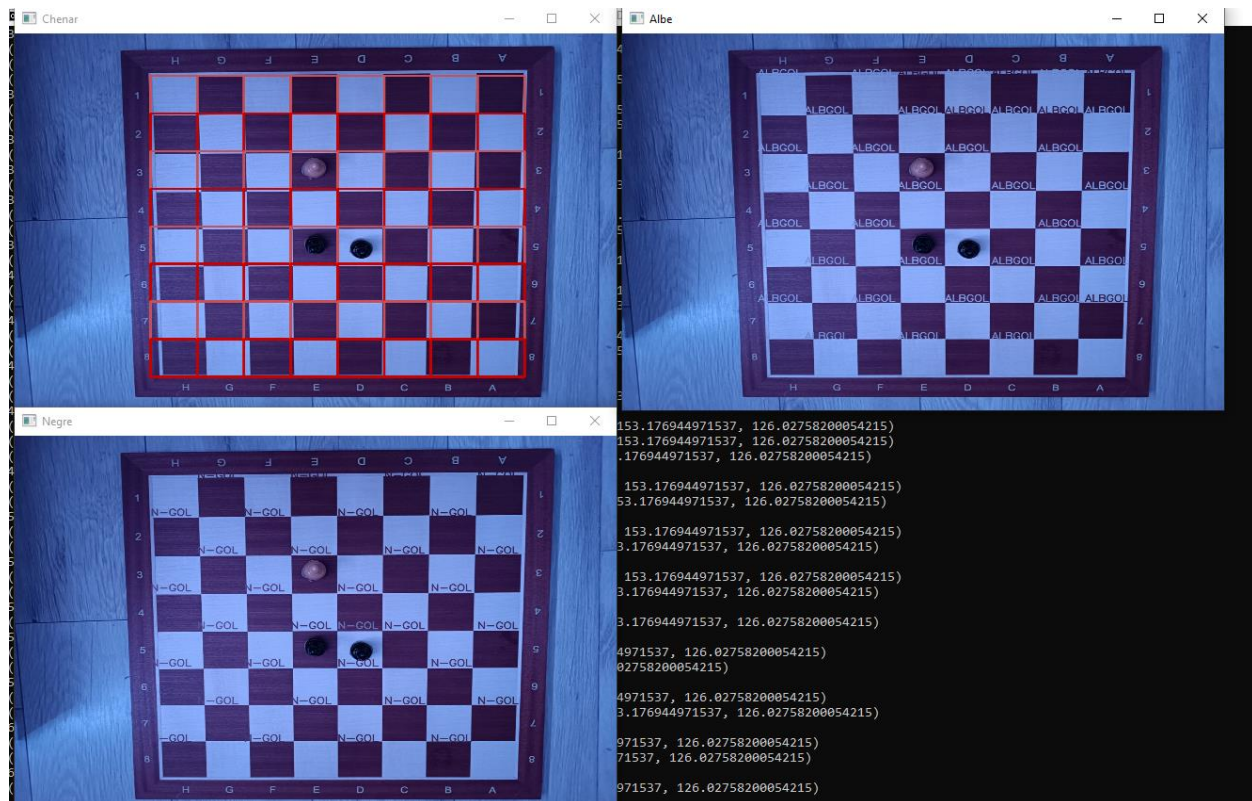


Figura 18 Rezultate obținute în urma algoritmului de mediere în lumină albastră

Experimentele în condiții de lumină ridicată îmbunătățesc rezultatele pentru detecția câmpurilor albe, însă pentru a găsi câmpurile negre libere rezultatele sunt neschimbate după acest algoritm propus.





*Figura 19 detecție în lumină naturală*

Este necesar sa se țină cont de cantitatea de lumină ci si factori precum compensarea mediei unui pătrat cu ajutorul unei piese de culoare opusă, dar și valoare inițială a piesei deși pentru detecția de câmpuri albe există o funcție de diferență care acceptă rezultate ce diferă cu până la 20% față de valoarea inițială, această pondere din urmă putând fi configurată ușor fără a pierde din calitatea rezultatelor în cazul câmpurilor negre valoarea medie a unui câmp negru este așa de scăzută și lipsită de diferențe încât orice piesă de șah se poate afla pe un câmp de culoare neagră poate oferi un fals pozitiv



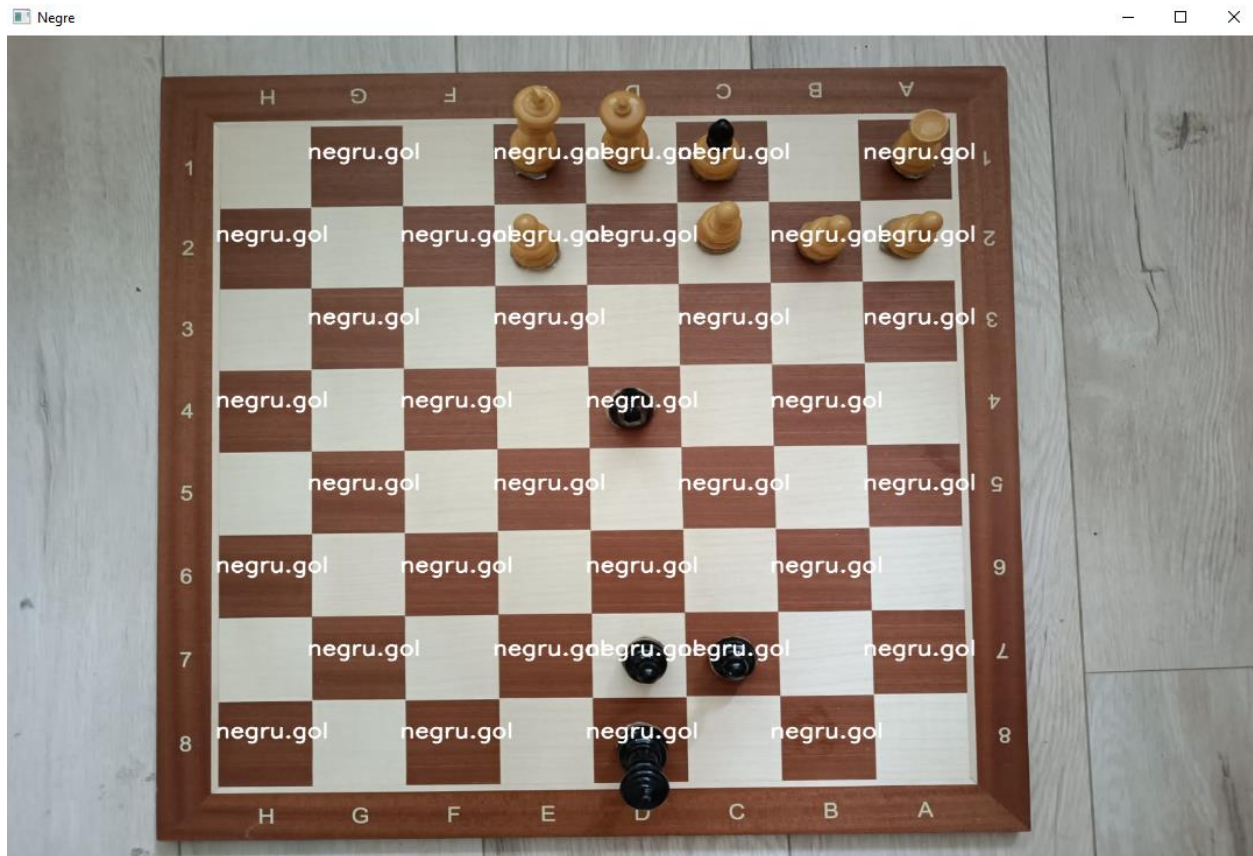


Figure 20 Functia de detectie a unui camp negru gol

#### 4.1.2 Generare de cod G

Pentru algoritmul de generare de cod se precizează parametri de configurație în antetul prestabilit iar mai apoi se scriu instrucțiunile succesiv. În general se va regenera câte un fișier nou pentru fiecare mutare. Fișier care mai apoi va fi trimis serial către placa Arduino care va executa o singură mutare, însă pentru testarea codului în diferite situații și mutări s-a dat algoritmului o succesiune de mutări care să implice diverse partide consacrate.

Pentru generarea de mutări care pleacă de pe un câmp interior tablei returnează rezultate bune, iar codul G este corespunzător, însă pentru câmpurile aflate la marginea tablei se generează o mutare ce va duce piesa în afara tablei din cauza algoritmului proiectat. Din această cauză se adugă o condiție care să simuleze un caz de integer

underflow, iar valorile odată ieșite din spațiul tablei să fie relocate cu ajutorul unei incrementări cu valoarea 8.

The image shows two Notepad windows side-by-side. The left window, titled 'mutari.txt - Notepad', contains a list of moves in algebraic notation: (e2,e4), (e7,e5), (b8,c6), (c8,f5), (b1,c5), (g8,f6), and (c1,f4). The right window, titled 'MutareGenerata.txt - Notepad', shows the corresponding G-code generated from these moves. The G-code includes standard settings like G21 (millimeters), G17 (XY plane), G94 (units per minute), and M3 (spindle on). It then uses G0 to move to absolute coordinates and G1 to move to relative coordinates for each move, with a feed rate of 90 (F90) and a dwell time of 5 (P5) between moves.

```

mutari.txt - Notepad
File Edit Format View Help
(e2,e4)
(e7,e5)
(b8,c6)
(c8,f5)
(b1,c5)
(g8,f6)
(c1,f4)

MutareGenerata.txt - Notepad
File Edit Format View Help
G21 ; millimeters G90 ; absolute coordinate
G17 ; XY plane
G94 ; units per minute feed rate mode
M3 S1000 ; Turning on spindle

; Go to zero location
G0 X0 Y0;('e2', 'e4')
G1 X3 Y4
m5
g4 p1
G01 X0 Y-2
m3 s90
G1 X3 Y4
g4 p1

;('e7', 'e5')
G1 X3 Y5
m5
g4 p1
G01 X0 Y2
m3 s90
G1 X3 Y5
g4 p1

;('b8', 'c6')
G1 X5 Y6
m5
g4 p1
G01 X1 Y2
m3 s90
G1 X5 Y6
g4 p1

;('c8', 'f5')
G1 X2 Y5

```

Figure 21 Generare de cod G pentru o succesiune de mutări

#### 4.1.3 Testarea sistemului mecanic

Sistemul mecanic a fost supus unui test pentru un scenariu de joc predefinit, care constă în primele 4 mutări ale partidei italiene: 1. e4 e5, 2. Nc4 Nc5, 3. Nf3 Nf6, 4. O-O O-O.

Prima încercare a constat în deplasarea efectorului de la poziția H1 către prima piesă și continuarea în aceeași direcție. Cu toate acestea, această strategie s-a dovedit a fi ineficientă, deoarece majoritatea mutărilor se desfășoară în general în centrul tablei sau în centrul extins (conturatul din jurul căsuțelor e4, e5, d4, d5). Cu toate acestea, aproximativ 80% dintre mutări au fost realizate cu succes, iar principalele eșecuri s-au produs la mutarea cailor, deoarece jumătățile de pas au fost pierdute în urma alunecării sau frecării efectorului pe tabla de șah sau pe suportul inferior. De asemenea, o mutare mai complexă, precum rocada, a necesitat adăugarea suplimentară a unui magnet și construirea unui traseu special, conform matricei de mutare a calului.

O altă problemă generată de frecare constă în executarea unor pași incompleți, ceea ce duce la devierea traiectoriei dorite. În cazul pieselor care se deplasează și pe linii între căsuțe, pot apărea coliziuni și decuplări ale pieselor.

Pentru a doua generație de teste, s-a adăugat o rolă cu flanșă în partea inferioară a modului de translație, iar poziția de plecare a efectorului a fost calculată de pe poziția centrală, respectiv de pe poziția E4. Mutarea punctului de plecare al prehensorului a redus timpul și distanța parcursă de acesta, iar rola a redus flambajele interne ale piesei de îmbinare a modulelor și a îmbunătățit cursa pe axa X.

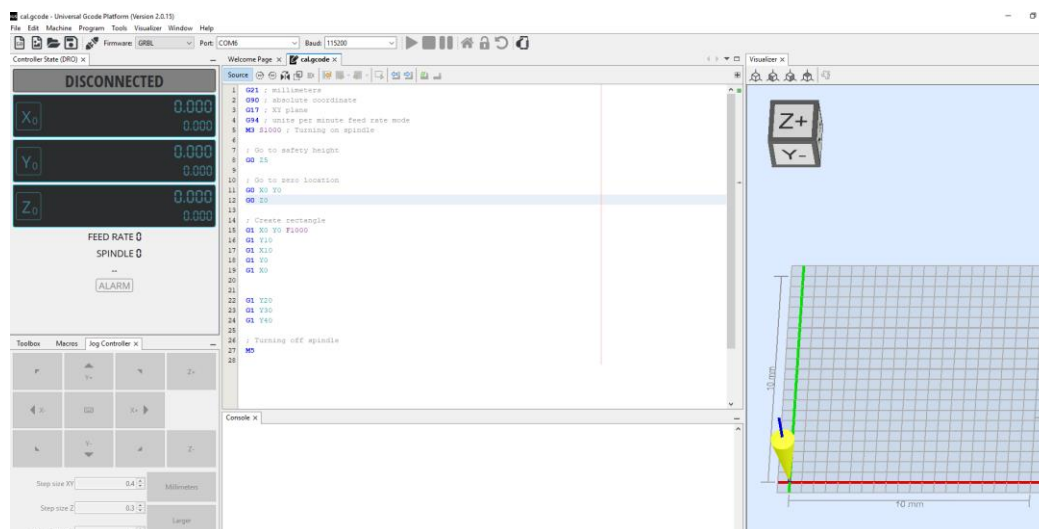
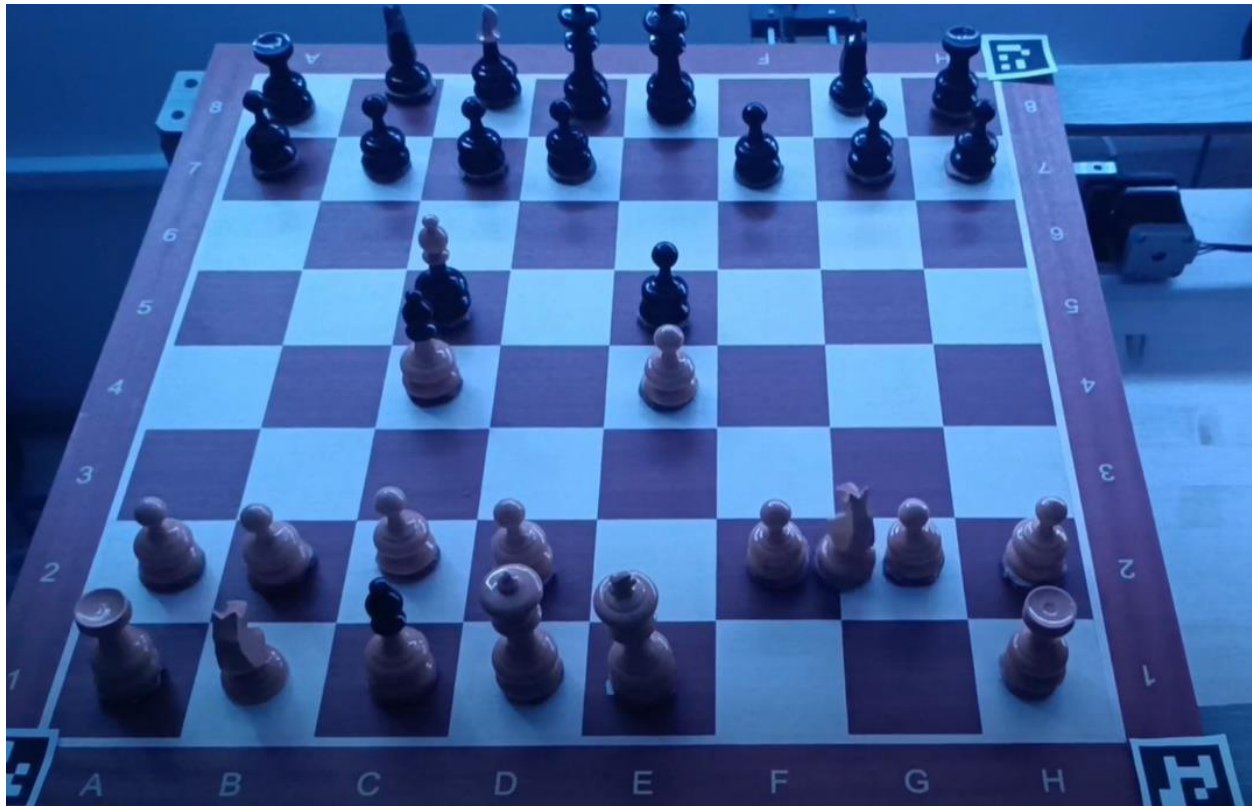


Figura 22 UGS aplicație de transmis cod G



*Figura 23 sistemul mecanic mutând calul*

#### 4.2 Utilitatea temei și direcții viitoare de dezvoltare

Robotul propus poate fi utilizat pentru antrenarea și exersarea jocului de șah la un nivel avansat într-un mediu familiar. În viitor, pot fi aduse mai multe îmbunătățiri, cum ar fi:

Extinderea tablei de șah: Se poate dezvolta o extensie a tablei de șah în care să poată fi plasate piesele capturate. Această extensie ar trebui să aibă suficient spațiu pentru a găzdui un număr maxim de 30 de piese de șah și ar trebui să permită o aranjare strategică astfel încât, la finalul jocului, piesele să poată fi rearanjate într-un număr minim de mutări.

Funcția de așezare automată a pieselor: La finalul jocului, prin apăsarea unui buton de reset, robotul ar trebui să aducă atât piesele proprii, cât și piesele adversarului înapoi în poziția inițială de joc într-un mod rapid, cu un număr minim de mutări.

Funcția de autocorectare a poziționării pieselor: Prin implementarea unei bucle închise în jurul centrului fiecărei celule a tablei de șah, camera poate înregistra poziția pieselor și, dacă un algoritm decide că poziția nu este centrală, pot fi generate mici mișcări în sens invers

pentru a atinge o poziționare exactă. Aceasta asigură o poziționare precisă și corectă a pieselor pe tabla de șah.

**Ceas de partidă:** Pentru a permite jocul conform regulilor celor 3 formate oficiale de șah (standard, rapid, blitz), se poate integra un ceas de partidă în robot, astfel încât jucătorii să poată gestiona timpul alocat fiecărei mutări.

**Interfață grafică și animație digitală:** Se poate dezvolta o interfață grafică intuitivă pentru utilizator, oferind mai multe opțiuni și o animație digitală în timp real pentru a urmări exact ceea ce vede robotul în timpul jocului.



*Figure 24 Ceas de Șah implementat cu arduino*

**Conexiune la internet și serviciu web:** Prin adăugarea unei conexiuni la internet și unui serviciu web, utilizatorii externi pot să se conecteze la o interfață grafică și să transmită mutări în timp real robotului. De asemenea, pot vedea tabla de șah originală prin intermediul camerei atașate, recreând astfel o experiență de realitate augmentată, cu un jucător uman la un capăt și robotul la celălalt capăt.

Aceste direcții de dezvoltare viitoare vor aduce îmbunătățiri semnificative în ceea ce privește funcționalitatea, interacțiunea și experiența utilizatorului în joc.

## Rezumat

---

Proiectul a avut ca scop dezvoltarea unui robot autonom capabil să joace șah la fel ca un om. A fost dorită implementarea unui sistem mecatronic care să extragă informații bazate pe imagini și să genereze și execute mutări în răspuns.

Constructiv, robotul este compus din 3 module de translație, o plăcuță de comandă Arduino Uno, o cameră web, o structură de suport și o unitate de calcul externă. Inițial, structura robotului a fost realizată în 3D, utilizând programul CATIA V5R21, simulând mișcările, iar apoi a fost realizată implementarea fizică. Astfel, au fost realizate teste pentru anumite elemente 3D înainte de a fi printate

După finalizarea sistemului hardware, s-a implementat sistemul electric al robotului, precum și sistemul de comandă. Cele 3 motoare de curent continuu sunt alimentate extern de o sursă de 12V și sunt comandate prin intermediul unui scut motor de tip CNC. Extragerea informațiilor din imagine a fost realizată cu ajutorul unei camere web și a unor algoritmi simpli de natură matematică care necesită o calibrare manuală. Principalele funcții ale sistemului software sunt identificarea căsuțelor goale, evaluarea poziției și generarea unui răspuns care este transmis către sistemul de control.

Etapă finală constă în asamblarea tuturor acestor componente și testarea continuă a sistemului mecatronic în vederea îmbunătățirilor ulterioare.

# Abstract

---

The task that initiated this project was to create an autonomous robot capable of playing chess just like a human. The implementation involved a mechatronic system for image-based information extraction, as well as the generation and execution of responsive moves.

From a construction standpoint, the robot consists of 3 translation modules, an Arduino Uno control board, a webcam, a supporting structure, and an external computing unit.

Initially, the robot's structure was designed in 3D using the CATIA V5R21 software, simulating its movements before the physical implementation. This allowed for testing of certain 3D elements before they were printed.

After completing the hardware system, the robot's electrical system and control system were implemented. The three DC motors are externally powered by a 12V source and controlled through a CNC motor shield.

Information extraction from images was achieved using a webcam and simple mathematical algorithms that required manual calibration. The main functions of the software system include identifying empty squares, evaluating positions, and generating a response to be transmitted to the control system.

The final stage involves assembling all these components and continuously testing the mechatronic system for further improvements.

## 5.Bibliografie

- [1] Paolacci, G., & Chandler, J. (2014). Inside the Turk: Understanding Mechanical Turk as a participant pool. *Current directions in psychological science*, 23(3), 184-188.
- [2] The weird and wonderful history of chess-playing robots arduino blog
- [3] Newborn, M. (1997). Deep Blue and Garry Kasparov in Philadelphia. In *Kasparov versus Deep Blue: Computer Chess Comes of Age* (pp. 235-278). New York, NY: Springer New York.
- [4] Kołosowski, P., Wolniakowski, A., & Miatliuk, K. (2020, July). Collaborative robot system for playing chess. In *2020 International Conference Mechatronic Systems and Materials (MSM)* (pp. 1-6). IEEE.
- [5] Sušac, F., Aleksi, I., & Hocenski, Ž. (2017, May). Digital chess board based on array of Hall-Effect sensors. In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1011-1014). IEEE.
- [6] Frey, P. W. (1977). An introduction to computer chess. *Chess skill in man and machine*, 54-81.
- [7] Ramo, K. (2019). *Hands-On Java Deep Learning for Computer Vision: Implement machine learning and neural network methodologies to perform computer vision-related tasks*. Packt Publishing Ltd.
- [8] Bijl, P., & Tiet, A. P. (2021). *Exploring modern chess engine architectures*. Victoria University, Melbourne.
- [9] Dem'yanov, V. F., & Malozemov, V. N. (1990). *Introduction to minimax*. Courier Corporation.
- [8] Golz, J., & Biesenbach, R. (2015, November). Implementation of an autonomous chess playing industrial robot. In *2015 16th International Conference on Research and Education in Mechatronics (REM)* (pp. 53-56). IEEE.
- [9] Owen, G. (2013). *Game theory*. Emerald Group Publishing.
- [10] Knuth, D. E., & Moore, R. W. (1975). An analysis of alpha-beta pruning. *Artificial intelligence*, 6(4), 293-326.
- [11] Y., Cheng, L., Isenberg, T., Fu, C. W., Chen, G., Liu, H., ... & Wang, Y. (2021, May). Curve Complexity Heuristic KD-trees for Neighborhood-based Exploration of 3D Curves. In *Computer Graphics Forum* (Vol. 40, No. 2, pp. 461-474).
- [12] Lu, Y., Cheng, L., Isenberg, T., Fu, C. W., Chen, G., Liu, H., ... & Wang, Y. (2021, May). Curve Complexity Heuristic KD-trees for Neighborhood-based Exploration of 3D Curves. In *Computer Graphics Forum* (Vol. 40, No. 2, pp. 461-474).



- [13]Heinz, E. A. (1998). Extended futility pruning. ICGA Journal, 21(2), 75-83.
- [14]Hoki, K., & Muramatsu, M. (2012). Efficiency of three forward-pruning techniques in shogi: Futility pruning, null-move pruning, and Late Move Reduction (LMR). Entertainment Computing, 3(3), 51-57.
- [15] Sorin Grigorescu,Sisteme de Vedere Artificiala,Filtrarea Imaginilor
- [16]Shin, S. J., Suh, S. H., & Stroud, I. (2007). Reincarnation of G-code based part programs into STEP-NC for turning applications. Computer-Aided Design, 39(1), 1-16.

DECLARAȚIE PRIVIND ORIGINALITATEA  
LUCRĂRII DE LICENȚĂ / PROIECTULUI DE DIPLOMĂ / DISERTAȚIEI

UNIVERSITATEA TRANSILVANIA DIN BRAȘOV

FACULTATEA INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

PROGRAMUL DE STUDII .....Robotica.....

NUMELE ȘI PRENUMELE.....Paun Andrei.....

PROMOȚIA.....2019-2023.....

SESIUNEA .....Vara.....

TEMA LUCRĂRII / PROIECTULUI/ DISERTAȚIEI

.....Robosah:Robot asistent pentru jucat sah.....

CONDUCĂTOR ȘTIINȚIFIC .....Conf.Dr.Ing Cocias Tiberiu-Teodor.....

Declar pe propria răspundere că lucrarea de față este rezultatul muncii proprii, pe baza cercetărilor proprii și pe baza informațiilor obținute din surse care au fost citate și indicate conform normelor etice, în textul lucrării/proiectului, în note și în bibliografie.

Declar că nu s-a folosit în mod tacit sau ilegal munca altora și că nici o parte din teză/proiect nu încalcă drepturile de proprietate intelectuală ale altcuiva, persoană fizică sau juridică.

Declar că lucrarea/ proiectul nu a mai fost prezentat(ă) sub această formă vreunei instituții de învățământ superior în vederea obținerii unui grad sau titlu științific ori didactic.

În cazul constatării ulterioare a unor declarații false, voi suporta rigorile legii.

Data: 22/06/2023

Absolvent

Paun Andrei Ionut

(nume, prenume, semnătură)

