

Proiect la SSC

Nume: Cacaraza Tobias-Daniel

Grupa: 30232

Contents

1.Introduction.....	1
1.1 Context.....	1
1.2 Specifications.....	1
1.3 Objectives.....	2
2.Bibliographic study.....	2
3.Analysys.....	3
4.Design	5
5.Implementation.....	7
6.Testing and validation.....	8
7.Conclusions.....	9
8.Bibliography	10

I. Introducere

1.1 Propunerea de proiect: Program pentru comanda unui utilaj cu comanda numerica (utilaj implementat cu Arduino)

Cerinta: Se va scrie un program pentru o masina care misca un pix/creion. Programul citeste un fisier care contine traiectoria de scriere (secventa de segmente sau arce de cerc) si genereaza comenzi pentru deplasarea pe doua directii (x si y) a capului de scriere; deplasarea capului de scriere se va simula pe ecranul calculatorului. (Java, C, C++, C#, etc.).

1.2 Objective

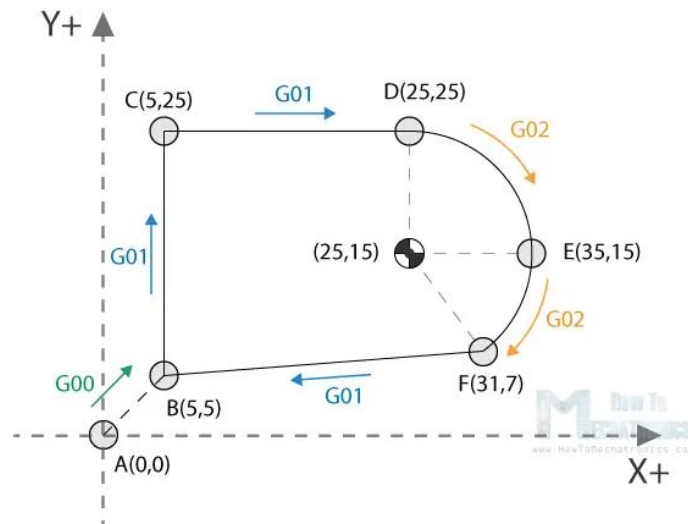
Sa gasesc aplicatiile de care am nevoie pentru proiect.

Cunoasterea limbajului G_code.

Dezvoltarea Programului: Crearea unui software usor de utilizat, capabil sa primeasca si sa interpreteze datele de intrare privind traiectoria de taiere si sa genereze comenzi precise pentru deplasarea capului de taiere pe doua axe (x și y).

Simularea deplasarilor.

Exemplu:



G00 X5 Y5 ; point B

G01 X0 Y20 F200 ; point C

G01 X20 Y0 ; point D

G02 X10 Y-10 I0 J-10 ; point E

G02 X-4 Y-8 I-10 J0 ; point F

G01 X-26 Y-2 ; point B

- Interfata simpla de folosit

1.3 Planificare

- Analiza cerintelor.
- Proiectarea si implementarea structurii: Crearea structurii programului, definirea claselor, functiilor și metodelor necesare pentru interpretarea datelor și simularea deplasării.
- Implementarea simulării deplasariilor.
- Optimizarea algoritmilor folositi, reducere de timp si resurse.

- Interfata de utilizator.

II. Bibliographic study

Interpolarea liniara este un algoritm simplu care aproximează valorile intermediare pe o traiectorie dreaptă între două puncte cunoscute. Algoritmul constă în calcularea punctelor intermediare în funcție de panta (inclinarea) dintre cele două puncte cunoscute.

Interpolarea curba : prin polinoame de interpolare Funcțiile de baza sunt polinoamele Lagrange

$$\varphi_k(x) = l_k(x) = \frac{\prod_{i=0, i \neq k}^n (x - x_i)}{\prod_{i=0, i \neq k}^n (x_k - x_i)},$$

Polinomul de interpolare este

$$g(x) = \sum_{k=0}^m c_k l_k(x).$$

$$l_k(x_j) = \begin{cases} 1 & \text{dacă } j = k, \\ 0 & \text{dacă } j \neq k. \end{cases}$$

III. Analiza

Pentru a controla mașini universale și alte echipamente tehnologice, sisteme de control numeric(CNC). CNC controlează mișcarea corpurilor de lucru ale mașinilor-unelte și echipamentelor, viteza acestora în timpul modelării pieselor, mișcările de instalare, precum și succesiunea modurilor de prelucrare și a funcțiilor auxiliare.

1. Structura fizică:

- Baza: O baza solida sau cadru pe care este montata întreaga masina.
- Componenta de miscare X: un motor pas-cu-pas montat pe axa X care permite miscarea in plan orizontal.
- Componenta de mișcare Y: Similar, un motor montat pe axa Y pentru miscarea în plan vertical.
- Cap de imprimare: un creion/pix care se deplaseaza pe axele X și Y pentru a desena sau taia.

- Unitatea de control: Aparatul Arduino sau controlerul CNC care controlează motoarele și mișcarea.

2. Interacțiuni și componente cheie:

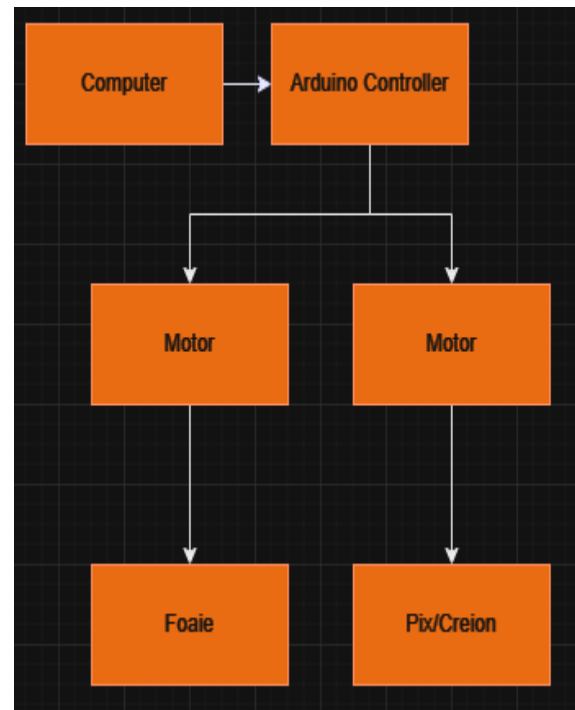
- Drivere motoare pas-cu-pas: Acestea asigură conectarea și funcționarea corectă a motoarelor
- Motoare pas-cu-pas: Acestea acționează componente de mișcare X și Y pentru a poziționa capul de imprimare.
- Controler Arduino: Unitatea de control care primește instrucțiuni și comenzi G-code și le traduce în mișcare fizică.
- Surse de alimentare: Sursa de alimentare sau bateriile care furnizează energie pentru motoare și controler.

IV. Proiectare

Diagrama de Interacțiune:

Diagrama de interacțiune arată cum componentele comunică între ele și cum fluxul datelor și comenzilor funcționează în cadrul sistemului. Acest lucru poate include:

- Trimiterea comenzilor G-code către Arduino de la un computer (interfata utilizator java application).
- Arduino procesează comenzile și coordonează motoarele.
- Motoarele furnizează feedback Arduino-ului cu privire la poziția actuală.
- Arduino interpretează comenzi, mișcă motoarele și comandă capul de imprimare.

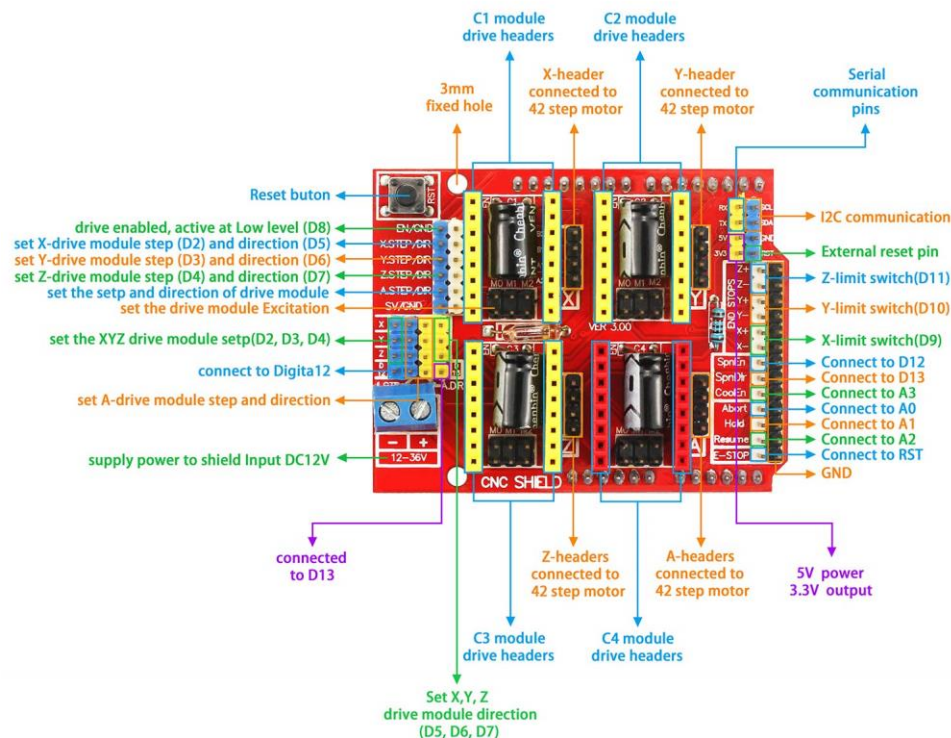


Computer Block:



Arhitectura:

Arhitectura unui mini plotter CNC Arduino poate fi conceputa ca un sistem modular, care include urmatoarele componente principale:



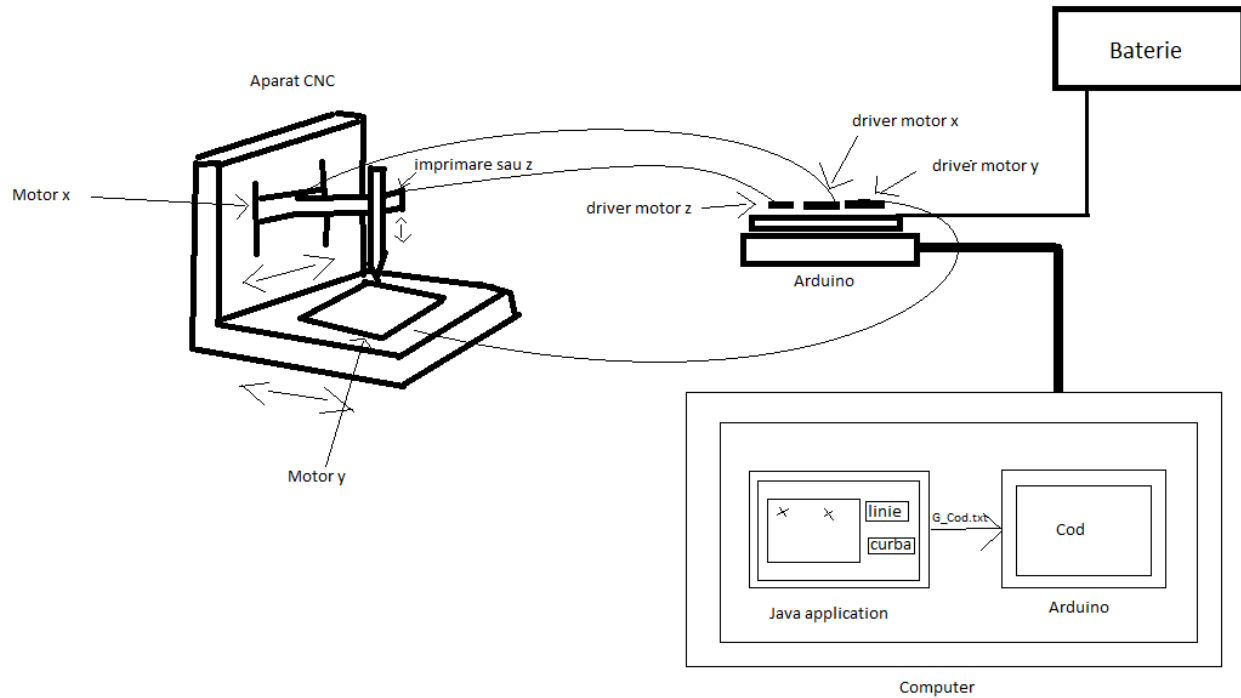
1. **Aplicatie pentru Desen (Interfata Utilizator):** Aceasta aplicatie Java are scopul de a crea desene si de a genera codul G-code corespunzator.
2. **Controler Arduino:** Aceasta este unitatea centrala a sistemului care primește comenzile si g-code din surse externe și coordoneaza toate miscarile.
3. **Motoare Pas-cu-pas:** Acestea controleaza miscarea pe axele X, Y, Z ale mașinii. Motoarele pot fi conectate direct la Arduino sau pot folosi drivere speciale (motoarele sunt conectate la pini de pe placa CNC, prin care este conectat la placa Arduino prin 2,5 (la X header), primul motor si 3,6 al doilea motor (la Y header), servomotorul este conectat la Z limit).

4. **Cap de imprimare:** Creionul/Pix.

5. **Surse de Alimentare:** Furnizează energie (baterie sau sursa).

Schema:

Schema poate evidenția conexiunile și interacțiunile dintre componentele principale ale sistemului. Acesta poate include conexiuni electrice, cabluri, plăci de circuit, senzori și circuite de control.

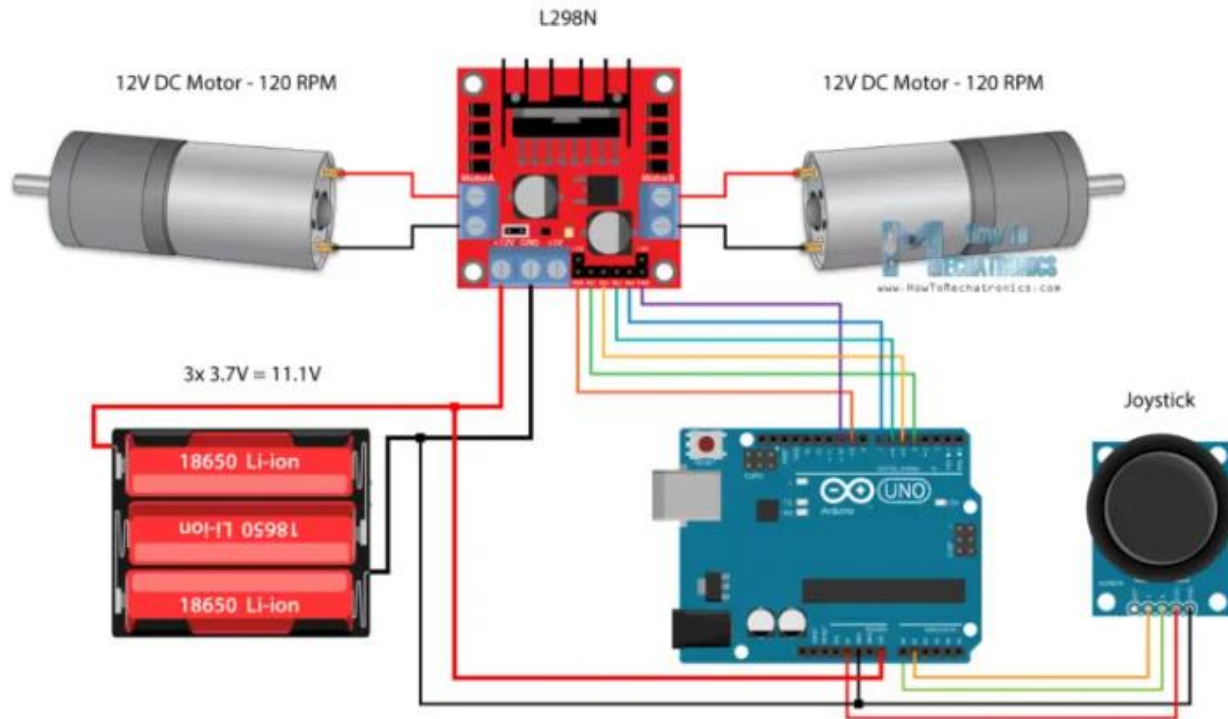


Modele de Date:

Modelele de date pot descrie cum sunt reprezentate și stocate informațiile în sistem. De exemplu:

- **Modelul de Trasare :** Cum sunt stocate și gestionate traiectoriile de desen sau tăiere, care pot fi reprezentate ca coordonate (x, y).
- **Modelul de Comenzi :** Cum sunt interpretate și procesate comenzile G-code primite de la surse externe (interfata Java) .

Exemplu de intrari si iesiri pentru Arduino:



În figura este un exemplu de legatură pini între componente.

V. Implementare

Codul Java este reprezentat de 2 interfețe care reprezintă conectarea cu placa Arduino și interfața de desenare. Conectarea cu placa Arduino / program Arduino am făcut-o prin thread-uri (un thread pentru citire de pe SerialMonitor a programului Arduino și un thread pentru interfața de desenare care trimite Gcodul în SerialMonitor). Pentru interfața de desenare, ca să fie mai ușor de folosit, la fiecare click am făcut să arate o bulină semnificând punctul de la care începe o linie/semicerc, urmând ca punctul viitor să fie sfârșitul liniei/semicerc (sunt desenate în timp real toate punctele). La începerea programului, utilizatorul desenează o linie, iar interfața are 3 butoane: linie, curbă și scriere Gcode. După apăsarea butonului de linie se va desena pe interfață linia, iar după apăsarea butonului de curbă va fi desenat semicerc, iar la apăsarea butonului de scriere Gcode se va trimite Gcodul mașinii Arduino. Algoritmii folosiți în desenare sunt algoritmul de interpolare liniară și algoritmul lui Bresenham luat din semestrul 2 din facultate (ECG).

Algoritmul de interpolare liniară (alg) Arduino

1 .Conversia coordonatelor:

Coordonatele spațiale inițiale (x_1 , y_1) sunt multiplicare cu factorii de conversie StepsPerMillimeterX și StepsPerMillimeterY pentru a obține numărul de pași necesari pe fiecare axă. Astfel, x_1 și y_1 devin numărul de pași necesari pe axele X și Y pentru a ajunge la poziția dorită.

2. Inițializarea poziției curente:

Se salvează poziția curentă a sistemului în variabilele x_0 și y_0 .

Calculul diferenței de coordonate:

Se calculează diferența dintre coordonatele finale (x_1 , y_1) și cele inițiale (x_0 , y_0) pe fiecare axă. Aceste diferențe sunt stocate în variabilele dx și dy .

3. Stabilirea direcției de deplasare:

Variabilele sx și sy sunt stabilite pentru a determina direcția de deplasare pe fiecare axă. Dacă x_0 este mai mic decât x_1 , atunci sx este setat la StepInc, altfel la -StepInc. Similar, pentru axa Y, sy este setat în funcție de relația dintre y_0 și y_1 .

4. Bucle de deplasare:

Se utilizează o buclă for pentru a realiza deplasarea pe axa cu cea mai mare diferență (dx sau dy). În fiecare iterație, se realizează un pas pe axa corespunzătoare (X sau Y), iar contorul over este actualizat.

5. Actualizarea coordonatelor:

Dacă over depășește valoarea corespunzătoare diferenței pe celălalt ax, atunci se realizează și un pas pe acest al doilea ax. Acest lucru asigură deplasarea sincronizată pe ambele axe, păstrând un traseu aproximativ liniar.

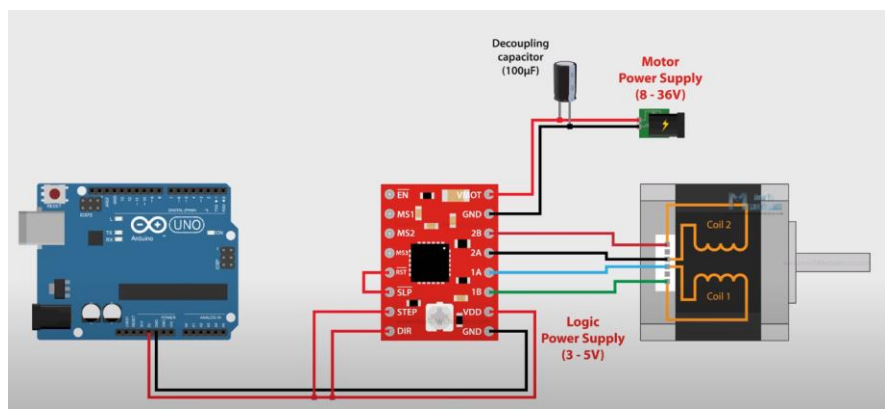
Algoritmul Brehensam calculează o optime dintr-un cerc și folosește același algoritm pentru toate cercurile ale cercului.

Codul Arduino reprezintă citirea de pe SerialMonitor a Gcodului trimis de către Java Maven. Gcodul are G0 care este o comandă mai rapidă folosită pentru a duce creionul/pixul la punctul de început sau la punctul mașinii de începere, iar G1 este folosit pentru scriere care folosește algoritmul de interpolare explicat la (alg) D și U care pune creionul/pixul pe foaie sau îl ridică cu comenzile M300 S100(D) și M300 S80(U) care fac același lucru, cu M100 care raportează poziția creionului.

VI. Testing and validation

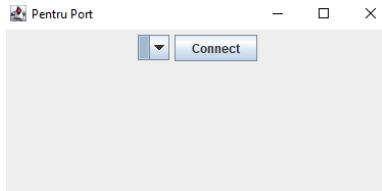
Testările au fost pe componente: Testarea motoarelor, drivere-lor și a servomotorului a fost prin cod Arduino și folosirea pinilor motoarelor [8].

Testarea conectării Arduino cu Java [9].



Testare a curbei si a liniei (fizic).

Testare conectare port



VII. Concluzii

Programul este destinat unei mașini care deplasează un pix sau un creion și simulează această mișcare pe ecranul calculatorului. Proiectul are multiple aspecte, inclusiv aspecte de programare în limbajul G-code, structura fizică a mașinii, componente cheie, analiză și implementare. Acest proiect oferă o abordare practică și hands-on asupra conceptelor din domeniul CNC, contribuind la dezvoltarea abilităților de proiectare, implementare și testare în context real.

VIII. Bibliografie

- [1] "G-Code Files: How to View, Edit, and Simulate - Best Software," MakeUseOf, [Online]. Available: <https://www.makeuseof.com/g-code-files-view-aimulate-best-software/>.
- [2] "Ultimaker Cura: Advanced 3D Printing Software, Made Accessible," Ultimaker, [Online]. Available: <https://ultimaker.com/software/ultimaker-cura/#links>.
- [3] "The Best G-Code Editors and Simulators (For Linux and Windows)," All3DP, [Online]. Available: <https://all3dp.com/2/best-g-code-editors-online-offline/>.
- [4] "G-Code Tutorial: List of Most Important G-Code Commands," HowToMechatronics, [Online]. Available: <https://howtomechatronics.com/tutorials/g-code-explained-list-of-most-important-g-code-commands/>.
- [5] "Diagram maker gleek_", aplicatie, [Online]. Available: https://app.gleek.io/diagrams/_my0mY4nIrlU5lDdb8bFOA.
- [6] "L298N Motor Driver – Arduino Interface, How It Works, Codes, Schematics", Dejan, [Online]. Available: on Arduino Tutorials

<https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>.

[7] "Metode de interpolare globala", Gabriela Ciuprina, [Online]

Available:

http://an.lmn.pub.ro/slides2017/04a_AN_handoutWithNotes.pdf.

[8] "Stepper Motors and Arduino – The Ultimate Guide", How to Mechatronics, [Online] youtube,

Available:

https://youtu.be/7spK_BkMJys?si=WNjFo8S7g_gdpzuT .

[9] "communication between Arduino and Java", bleau83, [Online],

Available:

<https://forum.arduino.cc/t/communication-between-arduino-and-java/592750> .