



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Estudi, disseny i desenvolupament d'una màquina CNC a petita escala, de tecnologia Arduino i amb operació planar i Visió Artificial.

Document:

Memòria final

Autor:

Marcel Teixé Ramon

Directora:

Rita Maria Planas Dangla

Titulació:

Grau en Enginyeria Electrònica Industrial i
Automàtica

Convocatòria:

Pròrroga, 2022.

ÍNDEX

ÍNDEX	1
IL·LUSTRACIONS I ON TROBAR-LES	3
ABSTRACT	5
1.INTRODUCCIÓ	6
2. ESTAT DE LA QÜESTIÓ	7
2.1 Màquines CNC	7
2.2 Mecanitzat amb raig làser.	9
2.2.1 Tall per làser	10
2.3 Control numèric informatitzat	11
2.3.1 Controlador (Arduino)	12
2.3.2 Drivers de control	15
2.3.3 Motors Pas a Pas	18
3. METODOLOGIA	20
4. PLANTEJAMENT DE LES SOLUCIONS.	22
5. DESENVOLUPAMENT DE LA SOLUCIÓ	25
5.1 Estructura general	25
5.2 Peces 3D.	26
5.3 Muntatge	34
5.4 Connexionat	40
5.5 Programa de visió artificial	43
5.5.1 Detecció de cantonades Harris	44
5.5.2 Algorisme d'ordenació dels punts d'interès	45
5.6 Programa de control del moviment	46
5.7 Protocol de comunicació	47
5.8 Programa de l'Arduino	48
6. Pressupost	49
7. CONCLUSIONS	50
	1

<i>7.1 Resultats</i>	50
<i>7.2 Millores</i>	53
8. BIBLIOGRAFIA	56

IL·LUSTRACIONS I ON TROBAR-LES		
NÚM. IL·LUSTRACIÓ	NOM	PÀG.
Il·lustració 1	Placa Arduino UNO	12
Il·lustració 2	Connexió entre placa Shield i Arduino	14
Il·lustració 3	Driver de control A4988	15
Il·lustració 4	Pinout driver	15
Il·lustració 5	Esquema de polsos del driver	16
Il·lustració 6	Taula de les configuracions de resolució del driver	16
Il·lustració 7	Característiques motor pas a pas	19
Il·lustració 8	Peces 3D impreses	26
Il·lustració 9	Forats per a subjectar els suports de l'aparell	34
Il·lustració 10	Suport de l'eix X amb casquets i rosca de llautó	34
Il·lustració 11	Suport eix Y amb rodament	34
Il·lustració 12	Suport eix Y amb motor i acoplador	35
Il·lustració 13	Suport collat a la base	35
Il·lustració 14	Suports Y i X amb varetes i eix	36
Il·lustració 15	Suport de motor de l'eix Z	36
Il·lustració 16	Rosca de llautó pel moviment en l'eix X	37
Il·lustració 17	Motor de l'eix Z	38
Il·lustració 18	Màquina amb tots els seus components muntats	38
Il·lustració 19	Aparell dins la seva cabina	39
Il·lustració 20	Esquema de la CNC Shield	40

Il·lustració 21	Esquema dels ponts per clonar eix	41
Il·lustració 22	Cablejat CNC Shield	42
Il·lustració 23	Detector de cantonades Harris	44
Il·lustració 24	Peces a replicar	50
Il·lustració 25	Resultat primera prova	51
Il·lustració 26	Resultat segona prova	51
Il·lustració 27	Lletra "A" amb unes cantonades assenyalades	54

1. ABSTRACT

El present treball final de carrera té com a objectiu el disseny i la creació d'una màquina de control numèric per computador, la qual vol simular una talladora làser per a plaques de ferro. La funció d'aquesta màquina és la de reproduir en 2D el contorn d'una peça donada. Aquesta no pot incloure línies corbes. Totes les parts de la màquina seran dissenyades de forma autònoma, és a dir, tant la part mecànica com els diferents programes que facin actuar els motors, per produir un moviment determinat, seran fets des de zero.

La màquina està formada per parts dissenyades per ser impreses en 3D a les quals s'hi han afegit el menor nombre de components externs per a poder ser funcional. El moviment es realitzarà mitjançant motors pas a pas i aquests estaran controlats per una targeta d'Arduino, uns drivers de control i la targeta CNC Shield. Les peces seran dissenyades en SolidWorks.

Un cop duts a terme tots els passos per a tenir l'aparell operatiu s'han fet un seguit de proves per a comprovar el seu funcionament. S'ha buscat un error menor al ± 1 mm, intentant que no hi hagi una penalització en la durada del procés. El resultat ha sigut que mentre la màquina només es movia en un eix no augmentava l'error, però al moure's en 2 eixos a la vegada i hi ha hagut problemes de sincronia entre els motors. Per acabar, s'han definit les possibles millores que es poden aplicar per aconseguir noves funcions o millorar les actuals.

The objective of this final project is the design and creation of a computer numerical control machine, which wants to simulate a laser carver for iron plates. The function of this machine is to reproduce in 2D the contour of a given piece. It cannot include curved lines. All parts of the machine will be designed autonomously, i.e., both the mechanical part and the different programs that make the motors act to produce a given movement will be made from scratch.

The machine is made up of parts designed to be 3D printed to which the smallest number of external components have been added in order to be functional. The movement will be done by stepper motors and these will be controlled by an Arduino board, some control drivers and the CNC Shield board. The parts will be designed in SolidWorks.

Once all the steps to have the device operational have been carried out, a series of tests have been done to check its operation. An error of less than ± 1 mm has been sought, trying not to have a penalty in the duration of the process. The result has been that while the machine only moved in one axis did not increase the error, but when moving in 2 axes at the same time and there have been problems of synchrony between the motors. Finally, the possible improvements that can be applied to achieve new functions or improve the current ones have been defined.

1.INTRODUCCIÓ

Després de l'aparició de Arduino l'any 2002, va emergir un fort moviment conscienciat amb les plataformes lliures, tant en programari com en maquinari. Aquest moviment juntament amb la filosofia del "Do It Yourself" va fer que sorgissin un gran nombre de projectes de totes les índoles, des de projectes simples, com a robots cartesianes, fins a projectes més seriosos, com pot ser una màquina de control numèric.

Tot això és possible, ja que juntament amb Arduino, han sorgit infinitat de complements i sensors associats, fàcils d'aconseguir, barats i que fan que gairebé qualsevol projecte sigui fàcil de dur a terme i a més econòmicament viable.

Un exemple ben clar i pròxim és el concurs de robòtica de la Universitat de Deusto. Aquest concurs té diverses categories, per a cadascú competeix un tipus de robot diferent, per exemple per a fer baralles de sumo o seguir una línia blanca en el menor temps possible. La plataforma d'Arduino ha aconseguit fer que persones sense estudis de robòtica poguessin realitzar tota mena de projectes. Per aquesta raó s'ha fet aquest projecte. Per poder demostrar que amb pocs recursos es pot aconseguir dur a terme un projecte tan aclaparador com pot ser la construcció d'una màquina de control numèric de manera autònoma en tots els seus apartats de desenvolupament.

2. ESTAT DE LA QÜESTIÓ

En aquest apartat es farà un resum de les diferents qüestions que afecten al projecte que es durà a terme. Com podem veure les màquines CNC han estat bastament emprades en la indústria per les facilitats que varen aportar respecte a la metodologia emprada anteriorment, la qual es basava en la mà d'obra i una maquinària mecànica poc desenvolupada.

Així doncs, les màquines CNC van suposar un salt qualitatiu per com es coneix la indústria fins al mitjans del s.XX, on es troba l'origen d'aquesta nova maquinària. Va ser John T. Parsons, enginyer aeronàutic, qui va desenvolupar per a primera vegada una màquina CNC. En aquest cas el seu ús era per a la indústria aeronàutica, ja que es necessitaven mecanitzats complexos i la interpolació d'eixos, difícils d'aconseguir amb la maquinària usada fins al moment. És amb les màquines CNC que s'aconsegueix realitzar superfícies complexes tridimensionals en una gran varietat de materials.

2.1 Màquines CNC

El concepte de control numèric es basa en l'automatització de màquines de producció que són controlades per comandes prèviament programades en un mitjà d'emmagatzematge. Les sigles CNC fan referència a Control Numèric per Computació; bàsicament, són màquines capaces de realitzar un gran nombre de tasques, que en l'actualitat, han estat el fonament bàsic dels sistemes robòtics més avançats. Aquest tipus de màquines han estat la clau en el desenvolupament industrial, ja que, molts dels processos que han portat a la modernització en la indústria actual, ha estat principalment per la implementació d'aquesta mena de maquinària en els processos, que requereixen una precisió i complexitat determinada.

Les principals avantatges que presenta el control numèric enfront dels mètodes convencionals de control de màquines són:

1. Permeten augmentar la flexibilitat de l'operació, així com la capacitat de produir formes complexes amb millor precisió dimensional, tanmateix mostren una bona repetibilitat, reducció de peces defectuoses, altes velocitats de producció, major productivitat i millora de la qualitat dels productes.
2. Es facilita l'ajust de les màquines.
3. Es poden realitzar més operacions amb cada configuració i el temps requerit per a l'inici i màquinat és menor en comparació amb els mètodes convencionals.
4. Es poden elaborar programes amb rapidesa i transferir-los a qualsevol computador amb el mateix programari.
5. Faciliten la feina de l'operador de la màquina ja que es requereix menys experiència i coneixement comparat amb un procés no automatitzat.

No obstant això, el control numèric també presenta algunes limitacions, les que més destaquen serien: el cost inicial relativament alt de l'equip, la necessitat i el cost del temps de programació i el manteniment especial requerit. Pel fet que aquestes màquines de sistemes complexos és important fer un manteniment preventiu per evitar qualsevol incident.

També cal esmentar, el desenvolupament que s'ha dut a terme en els sistemes CAD. Aquest tipus de sistemes emprenen un procés mitjançant ordinador per a millorar el disseny i la fabricació d'un producte en concret. Es realitzen de forma més ràpida, precisa i amb un cost menor sempre i quan s'apliqui la tecnologia informàtica adequada.

Aquest tipus de sistemes, ofereixen uns avantatges notables enfront dels mètodes tradicionals de disseny, ja que, mitjançant una computadora, es pot dissenyar una peça i analitzar les seves característiques com podrien ser: el contorn, forma i grandària una vegada ja emmagatzemats els dissenys bidimensionals o tridimensionals. Una vegada realitzat aquest procés, l'operador, pot modificar i manipular el disseny que s'havia fet prèviament amb major facilitat i així accelerar en el desenvolupament del producte en qüestió.

2.2 Mecanitzat amb raig làser.

El mecanitzat amb raig làser és una forma de màquinatge que utilitza la calor dirigida amb un raig làser. Aquest procés utilitza energia tèrmica per eliminar el material de superfícies metàl·liques i no metàl·liques. L'alta freqüència de la llum monocromàtica incideix a la superfície i fa escalfar-la ràpidament fins que el material es fon o es vaporitza. El mecanitzat amb raig làser és més adequat per a materials fràgils amb baixa conductivitat, però es pot utilitzar en la majoria de material.

Podem identificar diferents tipus de làser, a saber:

- Formats a partir de gasos com ara els làsers a base CO₂, Ar o He-Ne.
- Làsers d'estat sòlid. Són làsers excitats òpticament els quals el seu medi actiu és un cristall d'estat sòlid. Els més utilitzats són aquells que el seu medi ressonador és un rubí sintètic.
- Làser excimer, o làser exciplex, és una forma de làser ultraviolat que s'utilitza habitualment en la producció de dispositius electrònics, circuits integrats basats en semiconductors, cirurgia ocular i micromecanitzats.

El mecanitzat amb raig làser es pot utilitzar per a soldar, revestir, marcar, tractar superfícies, perforar i tallar, entre altres processos de fabricació. S'utilitza en les indústries de l'automòbil, naval, aeroespacial, electrònica i mèdica entre altres. De totes aquestes funcions amb la què s'aprofundirà és en el tall per làser.

2.2.1 Tall per làser

En 1965 es va utilitzar la primera màquina de tall per làser per a perforar forats en matrius de diamant. Aquesta màquina va ser fabricada pel Western Electric Engineering Research Center. Més endavant, a principis dels anys 70, aquesta tecnologia es va posar en producció per a tallar titani per a usos aeroespacials. Al mateix temps, els làsers de CO₂ es van adaptar per a tallar materials no metàl·lics, com els tèxtils, ja que, en aquell moment, els làsers de CO₂ no eren prou potents com per a superar la conductivitat tèrmica dels metalls.

La generació del raig làser implica l'estimulació d'un material emissor de làser mitjançant descàrregues elèctriques o llums dins d'un contenidor tancat. A mesura que s'estimula el material emissor, el feix es reflecteix internament per mitjà d'un mirall parcial, fins que s'aconsegueix l'energia suficient per a escapar com un flux de llum coherent monocromàtica. Normalment s'utilitzen miralls o fibres òptiques per a dirigir la llum coherent a una lent, que enfoca la llum en la zona de treball. La part més estreta del feix enfocat sol tenir un diàmetre inferior a 0,0125 polzades (0,32 mm). Depenent del gruix del material, és possible obtenir amples de tall tan petits com 0,004 polzades (0,10 mm). Per a poder començar a tallar des d'un altre lloc que no sigui la vora, es realitza una perforació abans de cada tall. La perforació sol implicar un raig làser premut d'alta potència que fa lentament un forat en el material, El temps per fer aquest forat triga diferent depenent del tipus de material i el seu gruix. Per exemple tarda entre 5 i 15 segons per a l'acer inoxidable de 13 mm de gruix.

Els raigs paral·lels de llum coherent procedents de la font làser acostumen tenir un diàmetre d'entre 1,5 i 2 mm (0,06-0,08 polzades). Aquest feix s'enfoca i intensifica normalment mitjançant una lent o un mirall fins a un punt molt petit d'aproximadament 0,025 mm, per a crear un feix làser molt intens. Per a aconseguir un acabat el més suau possible durant el tall de contorns, la direcció de la polarització del feix es fa girar a mesura que recorre la perifèria de la peça contornejada. Pel tall de xapa, la distància focal sol ser de 38 a 76 mm.

Els avantatges del tall per làser sobre el tall mecànic inclouen una major facilitat per a subjectar la peça, ja que no hi ha cap força mecànica que provoqui aquest tall una major precisió, ja que el raig làser és molt concentrat i no es desgasta durant el procés, hi ha menys possibilitats de deformar el material que s'està tallant, ja que els sistemes làser tenen una petita zona afectada per la calor i que hi han alguns materials que no permeten una bona mecanització utilitzant mètodes més tradicionals.

Per altra banda si comparem el tall per làser respecte el tall per plasma, el primer sol ser més precís i utilitzar menys energia en tallar; no obstant això, la majoria dels làsers industrials no poden tallar un gruix igual d'ample de metall que el plasma.

Les noves màquines làser que funcionen a major potència (6.000 watts, enfront dels 1.500 watts de les primeres màquines de tall per làser) s'acosten a les màquines de plasma en la seva capacitat per a tallar materials gruixuts, però el cost de capital d'aquestes màquines és molt major que el de les màquines de tall per plasma capaços de tallar materials gruixuts com la xapa d'acer.

2.3 Control numèric informatitzat

El concepte de control numèric es basa en l'automatització de màquines eina que són controlades per comandes prèviament programats en un mitjà d'emmagatzematge. Les sigles CNC fan referència a Control Numèric Informatitzat. Aquestes màquines són, bàsicament, màquines capaces de realitzar un gran nombre de tasques de forma ràpida i precisa.

Aquest tipus de màquines han estat la clau en el desenvolupament industrial, ja que, molts dels processos que han portat a la modernització de la indústria fins a arribar a l'actual, han estat principalment implementats utilitzant aquesta mena de màquinaria en tot tipus de procediments, que han requerit una certa precisió i complexitat.

Els orígens de les màquines CNC es remunten als anys 50 i van ser desenvolupades per l'enginyer John T. Parsons. Aquestes van ser aplicades per primera vegada per la indústria aeronàutica militar, per a realitzar mecanitzats complexos i així aconseguir superfícies complexes tridimensionals en una gran varietat de materials.

A continuació, es detallen en tres punts el funcionament bàsic d'aquestes màquines:

-La diferència que existeix entre una màquina convencional i una de control numèric és que una computadora controla la velocitat dels motors (eixos) i la posició. A causa d'això, es poden aconseguir millors resultats sobre la base de figures tridimensionals complexes, cercles i línies que segueixin una diagonal.

-Són capaços de moure l'eina en coordinació amb els eixos de la màquina. Amb això s'aconsegueix que les trajectòries del feix de làser aconseguixin línies multidimensionals complexes en comparació amb els mètodes manuals de gravat.

-La interfície que governa ordinador/màquina, controla el moviment dels dos eixos i el làser. Prèviament programada, executa totes les ordenis per si sola, això implica, que l'operari d'aquesta no ha d'estar manejant-la, així es pot estalviar temps per a fer tasques d'un altre tipus.

També cal esmentar, el desenvolupament que s'ha dut a terme en els sistemes CAD (Computer Aided Design), Abans de la dècada de 1970, tots els dissenys i dibuixos es realitzaven manualment, la qual cosa requeria molt de temps. Per això, la introducció del disseny assistit per ordinador (CAD) ha suposat un canvi en la manera de produir el disseny de forma més efectiva.

El CAD és una aplicació de programari que utilitza la tecnologia informàtica en el procés de disseny per a produir espais bidimensionals (2D), corbes, superfícies i sòlids en objectes tridimensionals (3D). S'utilitza sovint en la construcció, la fabricació i altres indústries com el procés d'enginyeria per a produir el disseny conceptual i la disposició dels productes i definir la resistència i l'anàlisi dinàmica dels components de fabricació.

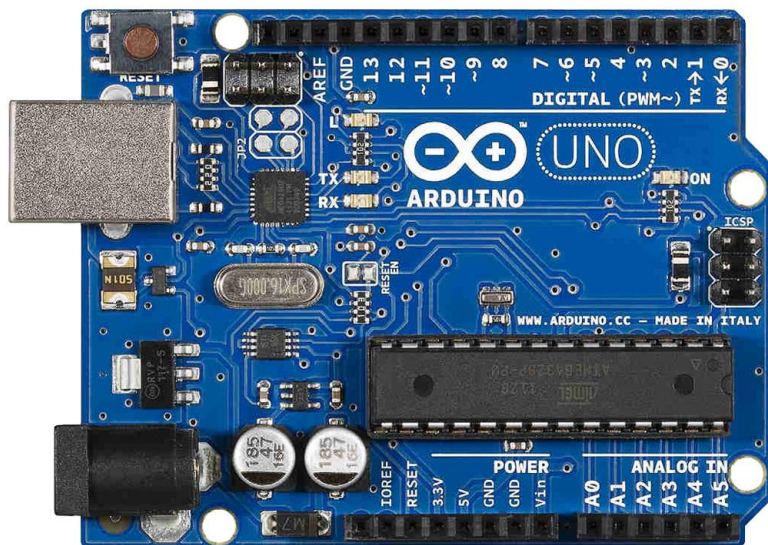
El CAD aporta avantatges als usuaris, però també comporta limitacions o inconvenients. Bàsicament, els principals avantatges són l'estalvi de costos i de temps, ja que el disseny i els plans poden realitzar-se més ràpidament que de manera manual. No obstant això, quan es produeix una falla tecnològica, especialment en l'ordinador, això afectarà la línia de producció i pot provocar una demora.

2.3.1 Controlador (Arduino)

El microcontrolador Arduino és una placa electrònica que s'usa bàsicament per a crear prototips, usa maquinaria i programari fàcil d'usar.

Es va crear a Itàlia l'any 2005 per un estudiant de l'institut IVRAE anomenat Massimo Banzi. El senyor Banzi va crear la marca Arduino amb la idea que tots els estudiants que estaven aprenent coneixements d'electrònica i computació tinguessin a l'abast una placa programable que fos assequible, tant en preu com en la seva programació. Des de llavors aquesta marca s'ha convertit en una plataforma líder mundial en tecnologia DIY (Do It Yourself).

Arduino utilitza un llenguatge de programació basat en Wiring i el seu desenvolupament està basat en Processing. Això permet que els projectes fets amb aquest dispositiu puguin ser executats sense necessitat de connectar-se a un ordinador, tot i que no deixa d'haver-hi la possibilitat de fer-ho i poder establir una comunicació amb diferents tipus de programari.



Il·lustració 1: Placa Arduino

Aquesta placa, pot ser construïda a mà (DIY) o bé comprada directament a la marca Arduino. El programari de control es pot descarregar de la web original i de manera gratuïta. Existeixen també fitxers pel disseny de referència que estan disponibles sota llicència Open Source.

En el mercat actual, existeixen molts tipus de plataformes amb microcontroladors, que realitzen funcions de computació física. Els més coneguts i utilitzats són: Raspberry Pi, BX 24 de Netmedia i Phidgets. Aquests ofereixen les mateixes funcionalitats que la plataforma Arduino per a programar un

microcontrolador però les plaques d'Arduino tenen certes avantatges. A continuació, es detallen en les característiques més rellevants enfront d'aquestes altres plataformes:

- Econòmica. la plataforma Arduino és dels pocs microcontroladors que ofereixen característiques tècniques molt completes a preus bastant assequibles.
- Compatibilitat amb diferents S.O., el programari IDE de Arduino és compatible amb els sistemes operatius més usuals: Windows, Macintosh i Linux. En comparació amb les altres plataformes de microcontroladors que només són funcionals amb Windows.
- Conta amb maquinari de codi obert disponible per als microcontroladors ATmega 168, ATmega 328 i ATmega 1280. Al ésser sota llicència de codi obert, els dissenyadors de circuits electrònics poden fer tasques d'optimització i fins i tot, millores en el disseny del dispositiu. Les llicències són sota Creative Commons.

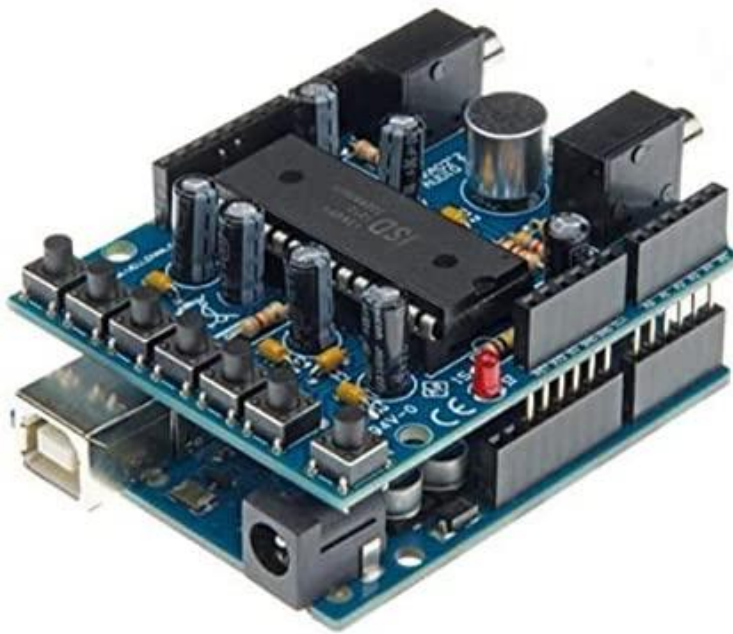
- Programació fàcil i senzilla, preparada per a principiants, estudiants d'enginyeria que vulguin complementar la seva formació en el camp de la programació de microcontroladors.
- En vista al programari del dispositiu és totalment de codi lliure. Això pot ser un al·licient per a programadors de nivell avançat que vulguin millorar i/o optimitzar la programació d'aquest microcontrolador. Està basat en el llenguatge AVR C, encara que es podria ampliar, mitjançant llibreries de llenguatge C++.

Dins dels microcontroladors de l'Arduino es diferencien tres tipus de memòria, cadascuna d'elles, compleix uns requisits determinats. Els tres tipus són: memòria Flaix, EEPROM i SRAM. Per a saber més sobre la mena de memòria de les anteriorment esmentades i com funcionen, es detallen a continuació, les seves principals funcions en l'Arduino.

- Memòria Flaix, es tracta d'una memòria que emmagatzema el programa en Arduino per a executar-lo. Quan es deixa d'alimentar la placa de Arduino, la memòria conserva el programa que aquesta porti. Depenent del microcontrolador que es vagi a utilitzar, l'espai d'emmagatzematge de la memòria variarà. Per al microcontrolador de l'Arduino UNO que utilitzarem en el nostre cas, és una memòria de 32 KB, usant-se 0.5 KB el bootloader d'aquesta memòria.
- EEPROM, per les seves sigles en anglès: Electrically Erasable Programmable Read-Only Memory. És una memòria tipus ROM programable, reprogramable elèctricament i esborrable, a diferència de la memòria EEPROM, ha de ser esborrada mitjançant raigs ultraviolats. Són memòries no volàtils, no necessiten energia per a conservar la informació dins d'ella. Pel microcontrolador que s'emprarà en aquest cas, ATmega328, té una grandària de memòria EEPROM d'1 KB. La memòria que hi ha a l'Arduino és petita, però admet afegir mòduls de memòria externa. Té una vida útil de 100.000 cicles, una vegada superats aquests cicles la memòria deixa de funcionar.
- SRAM, Static Random-Access Memory. És un tipus de memòria volàtil, és l'espai on els programes s'executen i manipulen variables, a diferència de les dues anteriors memòries, aquesta elimina la informació emmagatzemada quan és desproveïda d'alimentació elèctrica. És usada exclusivament quan el programa s'està executant. La grandària de memòria del xip que emprarem és de 2 KB.

Per a potenciar la funcionalitat de la plataforma és possible utilitzar Shields. Aquest tipus de dispositius es connecten a Arduino com a mostra en la següent il·lustració :

Aquests dispositius permeten ampliar les funcions de la placa d'Arduino. Les ampliacions més destacades utilitzant aquesta mena de dispositius són la de possibilitar la connexió



Il·lustració 2: Connexió entre placa Shield i Arduino

de manera contínua. Conté el programa amb les funcions principals.

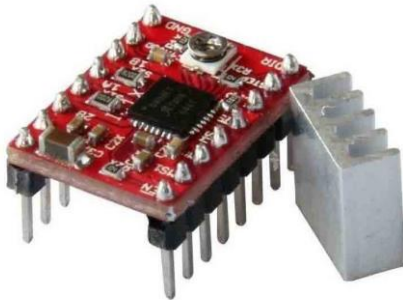
Per a poder inserir el programa en la placa s'ha de tenir instal·lat prèviament el IDE de l'Arduino, el qual, es pot descarregar de manera gratuïta en la pàgina oficial. La interfície inclou els drivers per a controlar la Shield, cosa que simplifica el procés de la seva instal·lació. Per a connectar el conjunt d'Arduino i Shield al ordinador, s'utilitza un cable USB que actua com a port serial.

amb cable ethernet, mòdul per a targetes SD, connexió Wi-Fi, drivers per a motors de corrent continua o pas a pas.

Com ja s'ha comentat anteriorment el llenguatge emprat en l'Arduino és una implementació de Wiring, basada en Processing, molt semblant al llenguatge C++. Per a poder executar el programa s'han d'emprar obligatòriament les dues següents funcions:

1. **Void Setup:** S'inicia una vegada quan s'inicia el sketch. Correspon a l'etapa a on es configura el programa i s'inicialitzen les variables,
2. **Void Loop:** Executat

2.3.2 Drivers de control

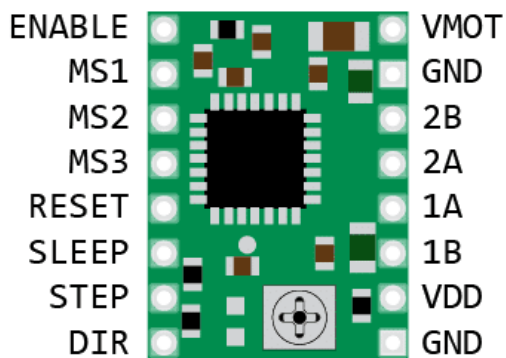


Il·lustració 3: Driver de control A4988

Per a l'elecció dels drivers que controlen els motors pas a pas de la màquina de control numèric, s'ha optat pels drivers Pololu A4988. S'han triat aquests drivers ja que són compatibles amb la placa Shield CNC. Tot i que existeixen altres tipus de drivers pel control d'aquesta mena de motors, s'ha considerat que aquests són els que millor s'adapten a l'objectiu del projecte que es durà a terme.

És un controlador molt senzill de manejar, serveix per a controlar motors pas a pas bipolars. Pot regular la intensitat de sortida mitjançant un potenciòmetre i suporta una intensitat de 2A per fase.

El dispositiu està dotat amb 16 pins els quals es poden configurar a gust de l'usuari, mentre els paràmetres de treball siguin els recomanats pel fabricant. Per a això, s'ha inclòs un esquema general del dispositiu i un breu resum de les funcions que pot realitzar cada pin.



Il·lustració 4: Pins driver

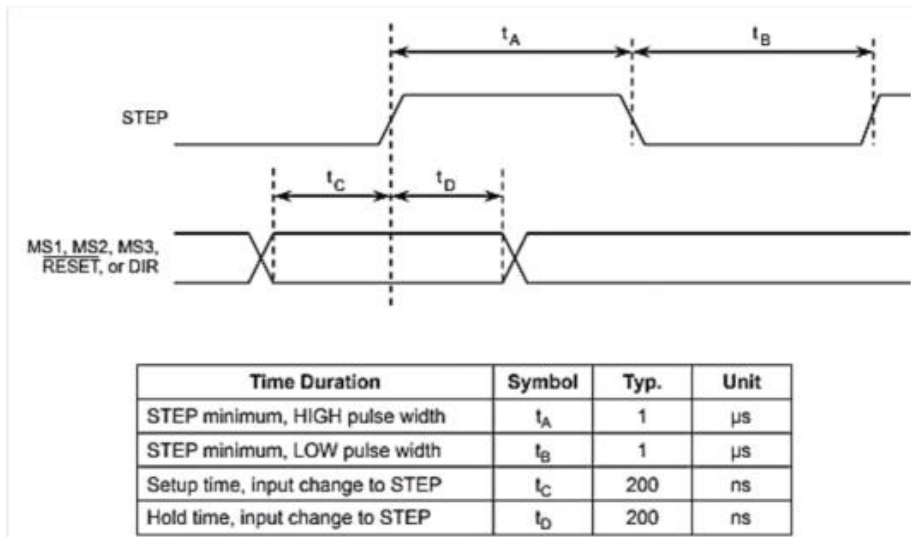
- **VMOT:** Alimentació (Motors). Voltatge admissible entre 7 V i 12 V.
- **GND:** Massa.
- **1A, 1B:** Entrades bobina 1 de motor.
- **2A, 2B:** Entrades bobina 2 de motor.
- **VDD:** Alimentació (Lògica). Voltatge admissible entre 3.3 V i 5 V.
- **DIR:** Direcció del Sentit de gir en l'eix del motor.
- **STEP:** Canvi de nivell baix a nivell alt,

l'eix del motor gira un pas.

- **SLEEP:** Manera activa a nivell baix, Driver entra en manera Standby.
- **RESET:** Manera activa a nivell baix. Entrades STEP deixen de funcionar, excepte si aquest pin es troba en manera nivell alt.
- **MS1, MS2, MS3:** Stepping (Passos de motor). Resolució.
- **ENABLE:** Driver en estat nivell alt, s'apaga. Sí l'entrada està connectada a massa, roman encès.

Encara que el Driver que hem triat pel projecte està dissenyat per a controlar motors bipolars hi ha la possibilitat de controlar motors unipolars. Interconnectant els conductors comuns de la connexió unipolar, podrem aconseguir un motor bipolar gràcies a aquest petit canvi.

Perquè el Driver funcioni correctament, es tindrà en compte el seu cicle de treball i el temps que han de funcionar els polsos. En la següent imatge extreta de la datasheet proporcionat pel fabricant, es pot observar la representació gràfica del cicle treball i el temps de durada de cada pols perquè el driver funcioni correctament.



Il·lustració 5: Esquema polsos driver

Per a configurar la resolució de passos que els nostres motors realitzessin és necessari col·locar uns jumpers o punts en els pins MS1, MS2 i MS3. Depenent a quin nivell de resolució es desitgi treballar, s'hauran de col·locar els punts com s'indica en la següent taula:

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase

Il·lustració 6: Taula de les configuracions de resolució del driver

En el nostre cas, com que volem una resolució lo suficientment petita per poder tenir un error petit però no volem ralentitzar en excés el procés, els pins MS1 i MS2 aniran puntejats i el pin MS3 no. D'aquesta manera, obtindrem una resolució de 1/8 passos cada vegada que rebi un pols de senyal elèctric.

En vista a la programació, per a poder programar un gir de motor pas a pas amb la resolució més gran mitjançant Arduino i que aquest envii l'ordre al Driver, s'ha emprat el següent Sketch:

```
#define STEP 6
#define RST 9
void setup () {
```

```
pinMode (STEP, OUTPUT); ;  
pinMode (RST, OUTPUT);  
digitalWrite (RST, LOW);  
}  
void loop () {  
for (int i=0; i<200; i++) {  
digitalWrite (STEP, LOW);  
delayMicroseconds (5);  
digitalWrite (STEP, HIGH);  
delay (1);  
}
```

2.3.3 Motors Pas a Pas

Per a dur a terme aquest projecte s'han utilitzat motors pas a pas model Nema 17.

Aquest tipus de motors electromecànics estan basats en desplaçament angular discret, és a dir, que a mesura que la seva entrada de control rebí un pols elèctric, aquest gira en un determinat angle. També s'han triat a causa del seu parell motor de 0.22N/m.

Existeixen altres tipus de màquines rotatives per a emprar en projectes similars. Els més emprats són: Motor Brushless i Motor de CC. Cadascun d'ells amb unes característiques tècniques determinades, per a saber més sobre aquests dos tipus de motors elèctrics s'ha fet un resum de les seves funcions i característiques principals.

- **Motor Brushless:** és un motor de corrent continu, la seva principal característica és que no posseeix escombretes en el rotor per a fer el canvi de polaritat. Per això, són motors que tenen un desgast mecànic pràcticament negligible.
- **Motor de corrent continu:** Aquest tipus de motors consta de dues parts, rotor i estator. L'estator suporta a l'eix i, al seu torn, conté el bobinatge principal de la màquina, ja sigui de pols permanents o bobinatge en coure sobre un nucli magnètic de ferro. El rotor té forma cilíndrica i té un bobinatge sobre nucli magnètic amb alimentació directa a les seves escombretes. L'inconvenient d'aquesta mena de motors és el seu manteniment i desgast per part de les escombretes ja que aquestes freguen sobre el col·lector de delgues.

Els motors per pas Nema 17 presenten unes característiques més adients pel nostre projecte, com per exemple: el manteniment és menys exhaustiu, les pèrdues per fricció són ocasionades pels rodaments, ja que no posseeixen escombretes, la precisió amb la qual són capaces de treballar és molt efectiva, la velocitat de gir és més fàcil de regular sense pèrdua de parell motor, posseeixen una millor resposta als senyals de control que se li envia i tenen una possibilitat de gir de 90° a petits moviments de centèsimes de grau.

Les característiques tècniques d'aquest motor es resumeixen en la següent taula proporcionada pel fabricant:

High Torque Hybrid Stepper Motor



Model		42SH33-1AM	42SH33-2AM	42SH33-3AM	42SH33-4AM
1 RATED VOLTAGE	V	4	6	12	2,8
2 CURRENT/PHASE	A	0,95	0,6	0,31	1,33
3 RESISTANCE/PHASE	Ω	4,2	10	38,5	2,1
4 INDUCTANCE/PHASE	MH	4	9,5	33	4,2
5 HOLDING TORQUE	NM	0,158	0,158	0,158	0,22
6 ROTOR INERTIA	G-CM ²	35	35	35	35
7 WEIGHT	KG	0,22	0,22	0,22	0,22
8 NUMBER OF LEADS	N ^o	6	6	6	4

II·lustració 7:Característiques motor pas a pas

El que s'ha escollit com a component de la nostra màquina ha estat el 42SH33-4AM.

3. METODOLOGIA

Per a tal de dur a terme aquest projecte, la metodologia ha passat des de la fase de plantejament i ideació de la màquina i el programa de control numèric; posteriorment per la fase d'obtenció dels materials i softwares necessaris per a poder dur a terme la tasca i finalment el muntatge d'aquesta i la creació del sistema de control numèric que traça el mateix sistema de moviment de la màquina.

Així doncs, podem distingir fins a 4 fases: plantejament i ideació, gestió material i ofimàtica, muntatge i disseny i l'última fase de testatge i conclusió.

FASE PLANTEJAMENT I IDEACIÓ

En aquesta fase es va plantejar la idea del maquinari, així mateix es va investigar l'ús d'aquestes màquines en l'àmbit industrial per a poder fer una reproducció a petita escala. Es va idear tant el tamany que es voldria aconseguir la màquina com tots els components, peces i cablejat necessari per a poder construir-la.

Més endavant, desenvoluparem els plànols que es varen dissenyar per a tal de poder preveure el material i les peces que serien necessàries per a poder tirar endavant el projecte.

FASE GESTIÓ DE MATERIAL I OFIMÀTICA

Per a poder obtenir els materials s'ha emprat una impressora 3D a través dels dissenys que es varen obtenir gràcies a la fase inicial. Un cop aconseguides les peces que es podien crear a través d'aquest sistema es van haver de localitzar proveïdors, ja siguin botigues físiques com de venda en línia dels matèries que eren precisos. En l'apartat de pressupost s'especifica amb més detall aquesta fase d'obtenció, així com els proveïdors usats.

FASE DE MUNTATGE I DISSENY

En un principi es van estar revisant diferents softwares que es podrien emprar en la fase del disseny de les formes i figures, així com el sistema de mapatge de la màquina per a poder reproduir aquests. Tot i que existeixen diferents softwares que podrien usar-se per a aquesta finalitat, finalment per aquest treball es va decantar per utilitzar SolidWorks per la familiaritat prèvia que es tenia en el seu ús, permeten agilitzar diverses parts de programació i ideació.

Gràcies a SolidWorks es va poder donar forma a diferents dissenys 3D que posteriorment la màquina hauria de poder reproduir tot tallant en alguna superfície plana i traslladant els eixos del disseny. Més endavant s'explica amb detall el sistema de mapatge que es va desenvolupar per a tal de què la màquina pogués produir els dissenys que li són donats.

No obstant això, la dificultat va ser poder crear un mapa que fos consistent tant pel programa com per la màquina i la seva extrapolació a 2D. Així doncs, el procés de depuració i testatge d'aquest va ser llarg i complex, ja que la solució de cada error o falla del sistema ens portava a nous reptes. Aquests detalls seran detallats amb més detall a l'apartat de desenvolupament de solucions.

La part de software va ser implementada i gestionada prèvia al muntatge de la màquina, ja que per tempos, mentre es produïen les peces per impressió 3D i s'aconseguia gestionar les que no es poden imprimir, es va aprofitar per fer el desenvolupament d'aquesta part més independent de la logística material.

El muntatge de la màquina es va a dur a terme en tres dies, els quals podríem resumir en el muntatge de la base i la cobertura de metacrilat, posteriorment es va realitzar el muntatge de la màquina amb els seus components, es va comprovar la seva funcionalitat mecànica i posteriorment es va introduir la part de cablejat i sincronització amb el programa SolidWorks.

FASE DE TEST I CONCLUSIÓ

Per últim pas, un cop ja es disposava de la màquina assemblada i el software maquejat per a connectar-se es va dur a terme la depuració de components externs (leds i botons) i de com actuaven els components uns sobre altres.

Un cop fetes les comprovacions es va poder iniciar la part més empírica del treball, posant a prova la màquina i realitzant diferents tests i comprovacions de la intercomunicació entre el programa de mapatge i la màquina. És en aquesta fase que s'han dut a terme varis prova-errors de diferents elements que ens han permès poder concloure el treball amb una anàlisi de resultats i les solucions aplicables a la màquina.

4. PLANTEJAMENT DE LES SOLUCIONS.

A l'estar implícit en aquest projecte que el programari a utilitzar per al desplaçament de la màquina serà l'Arduino en aquest apartat es farà un estudi de les diverses opcions per al que fa a la plataforma a fer servir per a desenvolupar el programa de visió artificial per a poder traduir la peça donada als diferents moviments a realitzar per poder-la reproduir.

Les plataformes les quals es realitzarà l'estudi són les següents: TensorFlow, OpenCV, SimpleCV i MATLAB.

TensorFlow

TensorFlow és una biblioteca de codi obert per a aprenentatge automàtic a través d'un rang de tasques amb una gran compatibilitat amb diversos llenguatges com ara C++ o JavaScript. Està desenvolupat per Google per a satisfer les seves necessitats de sistemes capaços de construir i entrenar xarxes neuronals per a detectar i desxifrar patrons i correlacions. Al tenir aquesta funcionalitat és obvi que la part de visió artificial té un gran pes en aquest programari.

Entre les seves principals virtuts hi ha el fet que es tracta d'una plataforma de codi obert la qual posa a la disposició totes les seves funcions als seus usuaris i que està preparada per al desenvolupament de qualsevol sistema. També proporciona una manera òptima de visualitzar les dades amb el seu enfocament gràfic i permet la fàcil depuració de nodes amb l'ajuda de la TensorBoard. Això fa que es redueixi l'esforç de visitar tot el codi i resoldre eficaçment qualsevol problema ocasionat.

El problema més gran que té aquesta plataforma és que està molt enfocada a usar-se en aparells amb sistema operatiu Linux i, tot i que sí que es pot utilitzar en Windows, no té les mateixes opcions ni està tan ben optimitzat.

OpenCV

OpenCV (Open Source Computer Vision Library) és una biblioteca de programari de visió per ordinador i aprenentatge automàtic de codi obert que es va crear per a proporcionar una infraestructura comuna per a les aplicacions de visió per ordinador i accelerar l'ús de la percepció automàtica en els productes comercials.

La biblioteca compta amb més de 2.500 algorismes optimitzats, que inclouen un ampli conjunt d'algorismes de visió per ordinador i aprenentatge automàtic. Aquests algorismes poden utilitzar-se per a detectar i reconèixer rostres, identificar objectes, classificar accions humanes en vídeos, seguir els moviments de la càmera, rastrejar objectes en moviment, extreure models 3D d'objectes, produir núvols de punts 3D a partir de càmeres estèrees, unir imatges per a produir una imatge d'alta resolució d'una escena completa, trobar imatges similars a partir d'una base de dades d'imatges, eliminar els ulls vermells de les imatges preses amb flaix, seguir els moviments dels ulls, reconèixer paisatges i establir marcadors per a superposar-los a la realitat augmentada, etc.

La biblioteca compta amb més de 47 mil persones de la comunitat d'usuaris i el nombre estimat de descàrregues supera els 18 milions. És per això que OpenCV es fa servir àmpliament en empreses, grups de recerca i organismes governamentals com per exemple a Google, Microsoft, Intel.

Una altra virtut OpenCV és que té interfícies en C++, Python, Java i MATLAB i és compatible amb Windows, Linux, Android i Mac OS.

Al contrari els seus desavantatges són una manca important en qualitat de la seva documentació, una corba de dificultat inicial elevada i el fet que la depuració i visualització dels diferents problemes pugui ser elevada.

MATLAB

La biblioteca de visió artificial de MATLAB, la Computer Vision Toolbox proporciona algorismes, funcions i aplicacions per a dissenyar i provar sistemes de visió per ordinador, visió 3D i processament de vídeo. Pot realitzar la detecció i el seguiment d'objectes, així com la detecció, extracció i correspondència de característiques.

També té la possibilitat d'entrenar detectors d'objectes personalitzats utilitzant algorismes d'aprenentatge profund i aprenentatge automàtic com YOLO v2, SSD i ACF. MATLAB permet accelerar els seus algorismes executant-los en processadors multinucli i GPU. Els algorismes de la caixa d'eines admeten la generació de codi C/C++ per a la seva integració amb el codi existent, la creació de prototips d'escriptori i la implantació de sistemes de visió integrats.

Una altra cosa que li dona un especial valor a aquesta llibreria és que compta amb una documentació excel·lent i una gran comunitat d'usuaris la qual permet una fàcil investigació per a resoldre qualsevol problema.

Tot i el seu gran nombre de punts positius, aquesta llibreria continua tenint inconvenients per al seu ús. El més gran sens dubte el fet que el programa MATLAB tingui un cost elevat per a obtenir la seva llicència. D'altra banda, també pot resultar complicat el seu llenguatge de programació si un no està acostumat a utilitzar-lo i acostuma a ser més lent que altres llenguatges.

SimpleCV

SimpleCV és una interfície per a biblioteques de visió artificial de codi obert en Python. Proporciona una interfície concisa i llegible per a les cambres, la manipulació d'imatges, l'extracció de característiques i la conversió de formats.

Com el seu nom indica, SimpleCV és una eina creada per a simplificar la visió artificial i que persones amb poca experiència en aquest tipus de programari pugui arribar a crear aplicacions per als seus projectes. Per aquesta raó SimpleCV està basat en programació Python, més fàcil de fer servir que altres llenguatges com ara C++.

Els seus desavantatges principals són la seva inflexibilitat a l'hora del llenguatge a programa, ja que només es pot fer en base Python, i que en buscar la màxima simplificació no té la profunditat que tenen altres opcions.

Amb tot el que s'ha exposat de les diferents opcions possibles, la que ha estat escollida ha sigut la llibreria OpenCV per a C++ per les següents raons:

-Possibilitat de programar en C++. A causa de la meua falta de familiaritat amb altres llenguatges de programació com ara Python, hi ha algunes de les opcions que han sigut descartades des d'un inici com és l'exemple del SimpleCV.

-Gran eficiència en el seu procés. La majoria d'opcions consumeixen una gran quantitat de recursos i fan que s'allargui el temps d'ús del programa. Amb OpenCV, a l'estar optimitzat en C++ no té aquest mateix problema.

-Codi obert. En ser un programari obert totalment gratuït fa que sigui molt més atractiu de fer servir en comptes d'altres opcions a les quals s'ha de pagar la llicència, com és el cas de MATLAB.

-Ús extensiu. De totes les possibilitats anteriorment mencionades, OpenCV és la més usada per programes amb una funcionalitat semblant a la que es busca amb aquest projecte. Per tant, pot arribar a ser més fàcil el fet de buscar informacions i formes de resoldre els problemes que vagin sortint.

5. DESENVOLUPAMENT DE LA SOLUCIÓ

Per al desenvolupament de la solució farem diferències entre el que és la part de hardware del que és la part de software.

5.1 Estructura general

El moviment de la màquina serà realitzat mitjançant el moviment d'eixos roscats, els quals seran rotats mitjançant els motors pas a pas. Per a sostenir tant els motors com els eixos s'han dissenyat un seguit de peces 3D.

Per a que les peces es mantinguin estables durant el moviment, en paral·lel als eixos, es muntaran dos varetes llises d'acer inoxidable per a cada eix. Per a tenir un moviment més suau entre les peces es farà servir casquets de boles allà on les varetes hagin de passar per l'interior d'un component.

Els motors transferiran el seu gir a l'eix corresponent ajudant-se d'uns acopladors de 5mm, que és el tamany del cilindre que sobresurt del motor, als 8mm de diàmetre dels eixos. Aquests acopladors tenen un seguit de cargols petits que permeten subjectar amb fermesa tant l'eix com el cilindre del motor. Per a permetre el gir dels eixos mantenint la seva posició s'ha instal·lat un rodament de boles a l'extrem del eix contrari al del motor. D'altra banda per a traduir el moviment rotacional de l'eix a un moviment translació s'utilitzarà una rosca de llautó. Aquesta es colla en la peça requerida per mitjà de 4 cargols de M3.

La màquina està a l'interior d'una cabina creada a partir d'una plataforma de fusta de 2cm com a base, i unes parets i tapa de metacrilat. El metacrilat està subjectat entre si i amb la base mitjançant escaires. Les mesures són 50x50cm de maquinària, dels quals l'espai útil són de 35x30cm. Per poder cobrir-ho tot la càpsula fa 75x65x35 cm amb espai suficient per poder posar les parts electròniques del sistema. La tapa, posada en la part superior de la càpsula, fa 50x50 i té unes frontisses i, juntament amb una nansa, permet obrir i accedir al seu interior de forma fàcil. Al utilitzar metacrilat transparent per a la seva construcció per seguretat s'utilitzaran sempre ulleres de protecció especials pel treball amb làsers.

Un cop exposada l'estructura general es passarà a explicar el disseny de les peces 3D.

5.2 Peces 3D.

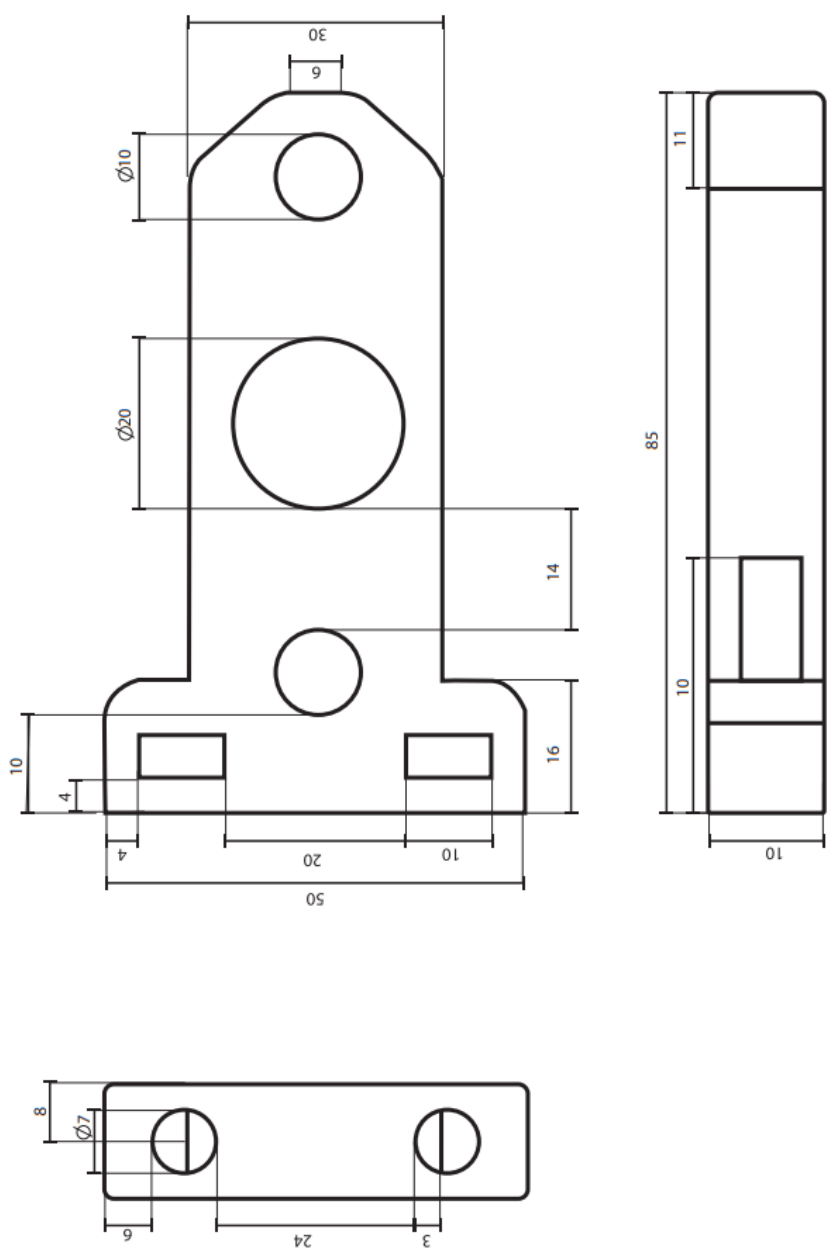
Pel disseny de les peces s'ha buscat que tinguin un disseny simple i funcional. Aquestes estan limitades per a la mida dels components extern ja que s'hi han d'inserir les varetes, collar els motors o introduir els casquets i els rodaments. En la següent taula es fa un llistat de les peces 3D.

DESCRIPCIÓ PEÇA	QUANTITAT
Suport motor eix X	1
Suport rodament eix X	1
Peça per la femella dels eixos roscats	2
Suport motor eix Y	2
Suport rodament eix Y	2
Suport del motor en eix Z	1
Suport làser	1
Acopladors dels casquets	4

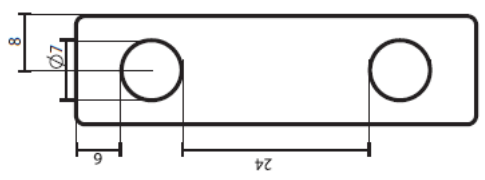
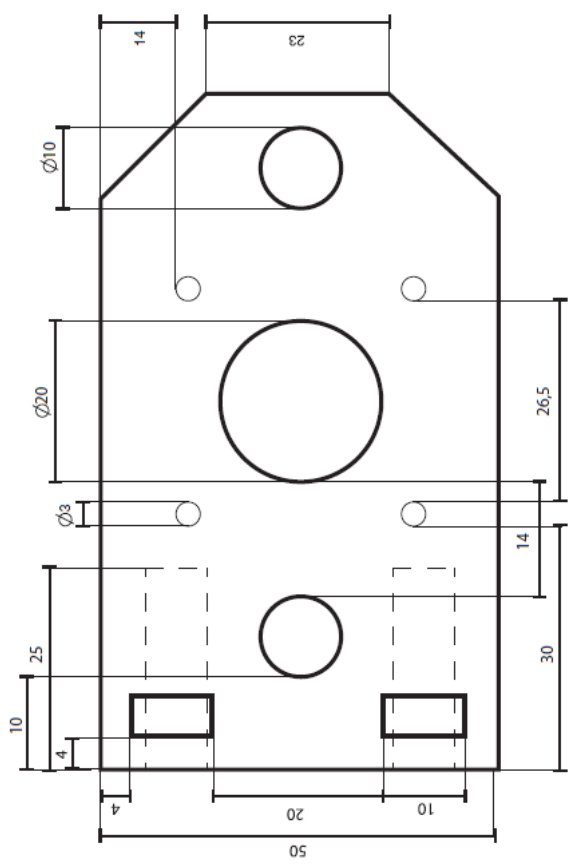


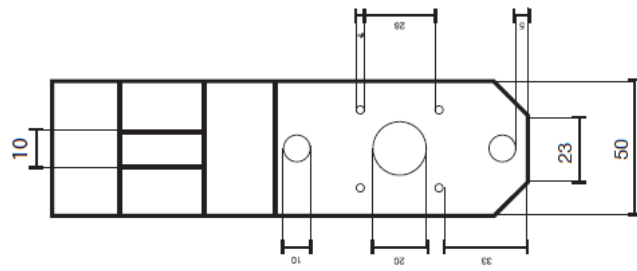
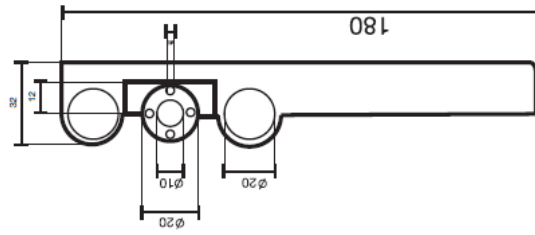
Il·lustració 8: Peces 3D impreses

Ara es passarà a mostrar els diferents plànols de cada peça.



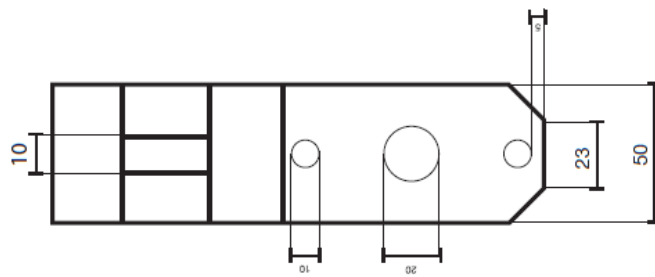
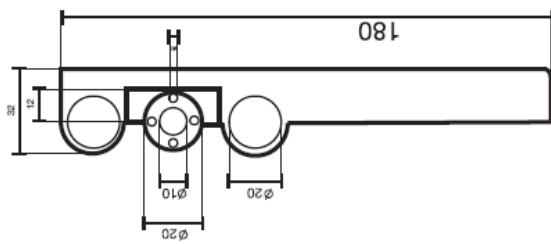
Títol: Suport Rodament en l'eix Y N°: 1

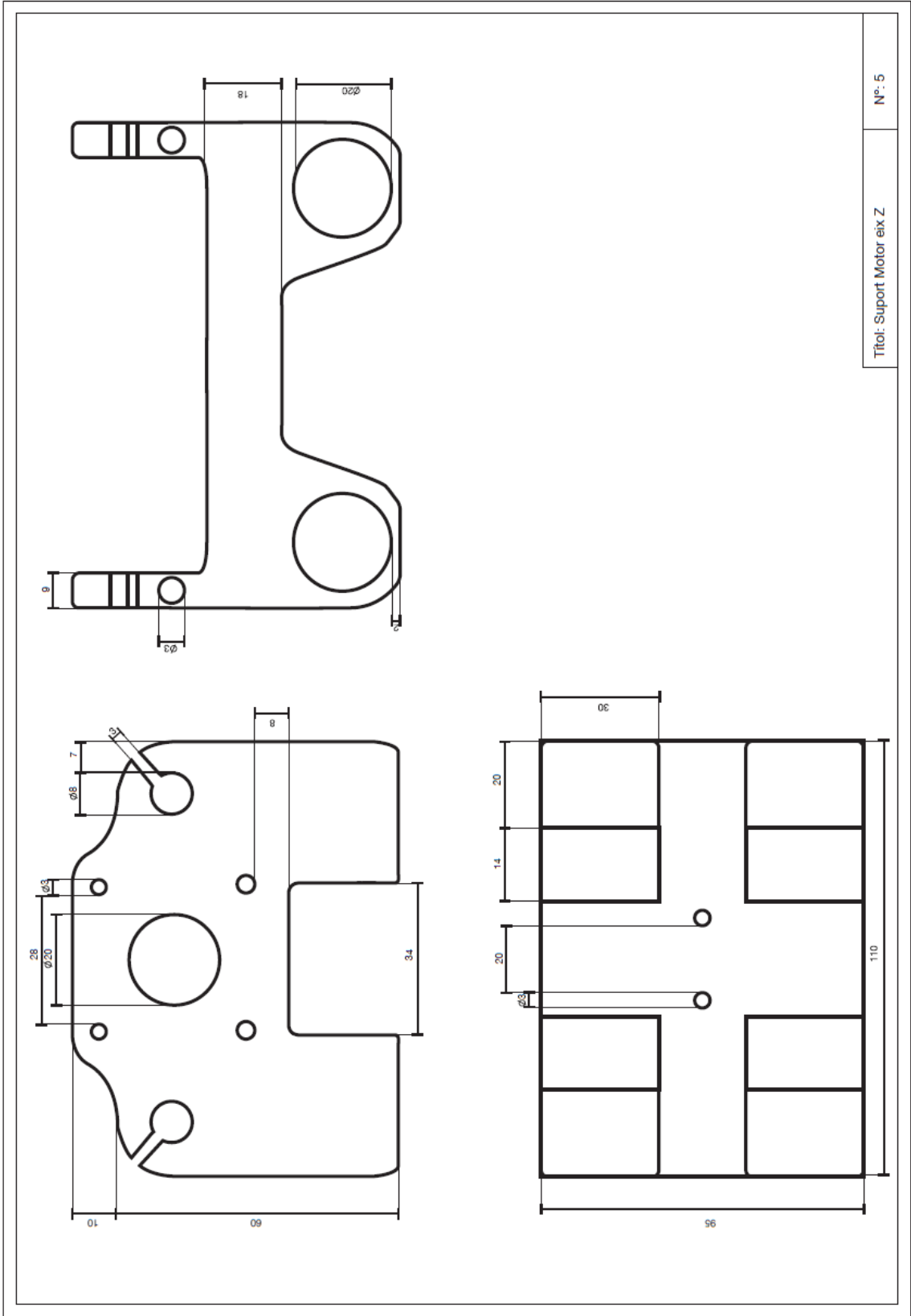




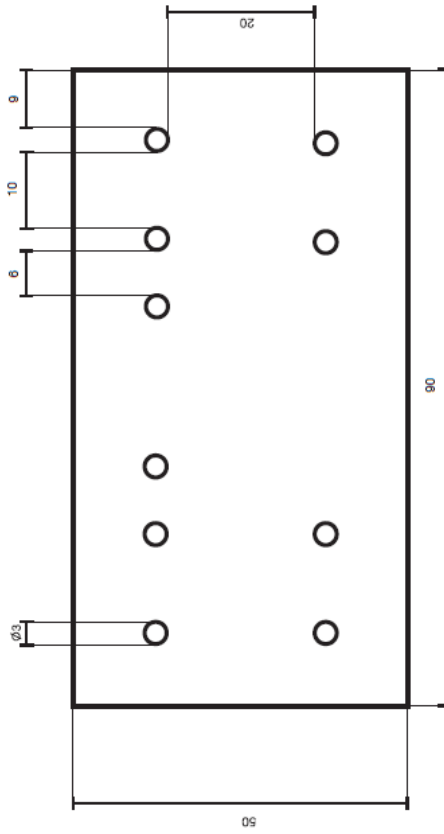
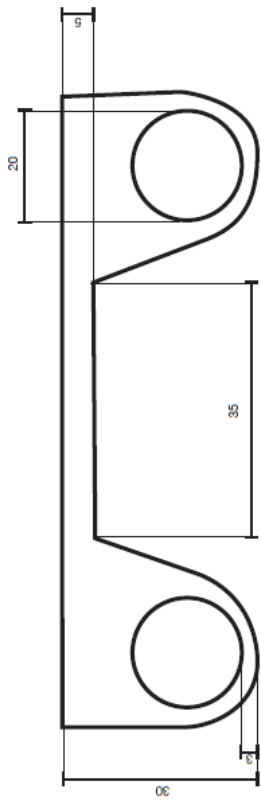
Titoli: Support Motor X

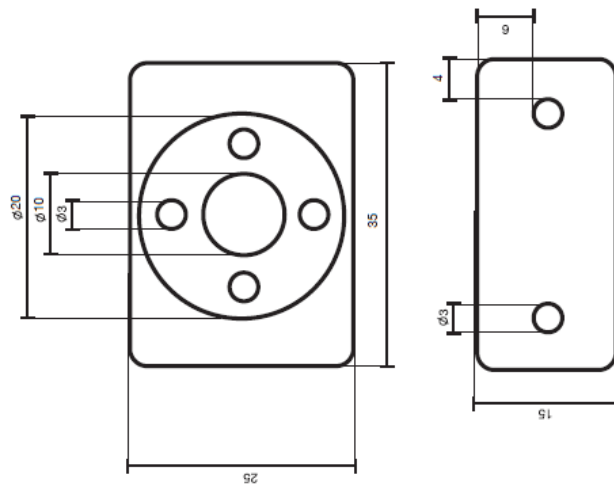
N°: 3





Titol: Suport Motor eix Z N°: 5





5.3 Muntatge

Per fer el muntatge de la màquina començarem marcant la fusta per poder fer els forats pels suports i els escaires. Un cop marcats passarem a foradar-los.



Il·lustració 9: Forats per a subjectar els suports de l'aparell

Un cop fet els forats passarem a muntar els casquets, els rodaments i les rosques de llautó en els seus llocs corresponents utilitzant els cargols necessaris.



Il·lustració 10: Suport del motor de l'eix X amb casquets i rosca de llautó



Il·lustració 11: Suport eix Y amb rodament

També aprofitarem per col·locar els acopladors als motors i atornillar aquests als suports corresponents.



Il·lustració 12: Suport de motor de l'eix Y amb motor i acoplador

Un cop fixats els elements externs als suports collarem les peces a la fusta. Posarem una femella en cada un dels forats rectangulars dels suports per poder subjectar la màquina de forma robusta.



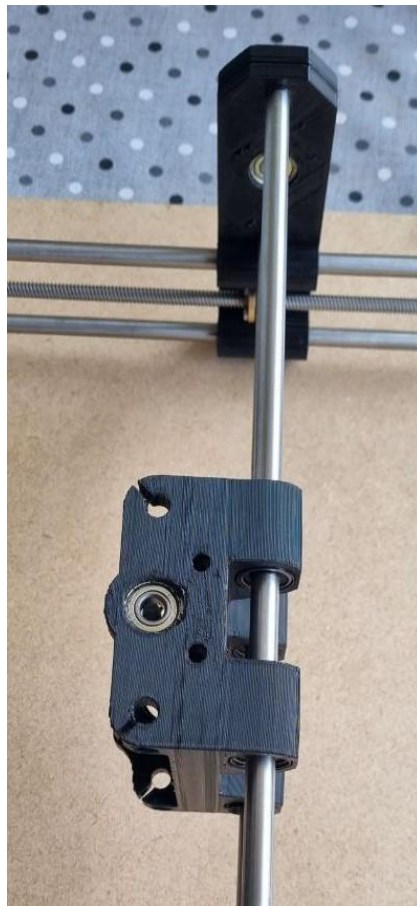
Il·lustració 13: Suport collat a la base

Després de fixar els peus de la màquina s'ha d'instal·lar les varetes llises i l'eix roscat juntament amb el suport encarregat del moviment en l'eix X.



Il·lustració 1410: Suports amb les varetes llises i l'eix roscat

Llavors posarem les varetes corresponents a aquest eix i al suport del motor de l'eix Z.



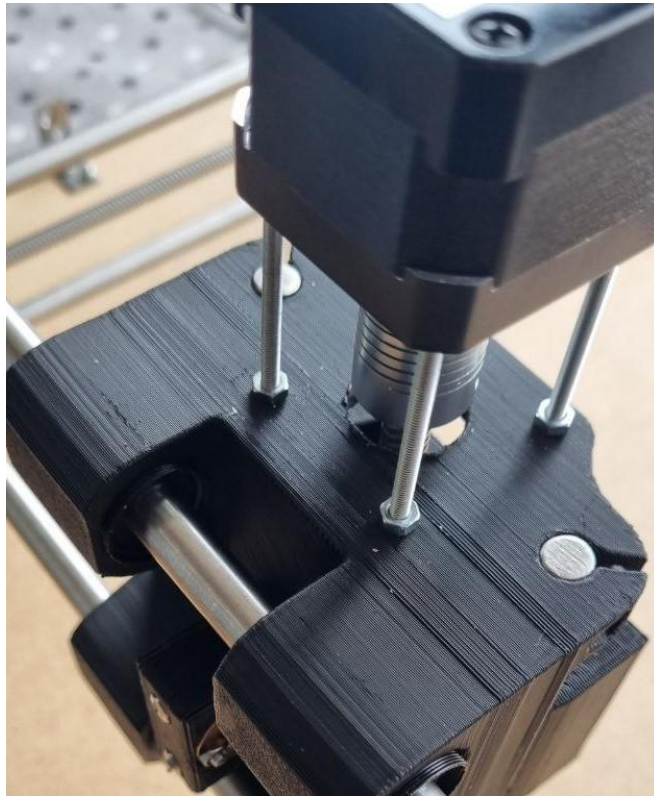
Il·lustració 115: Suport del motor de l'eix Z

També caragolem les peces on posarem les rosques de llautó que van tant a la peça del suport del motor en l'eix Z com al suport del làser.



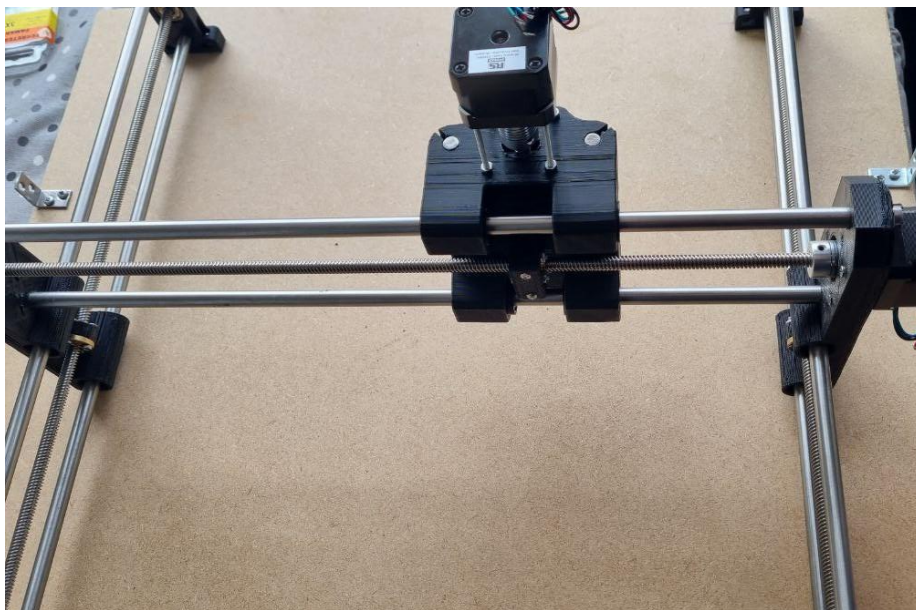
Il·lustració 16: Rosca de llautó de l'eix X

Una vegada muntat el suport, col·locarem el motor amb les varetes, l'eix i el suport pel làser.



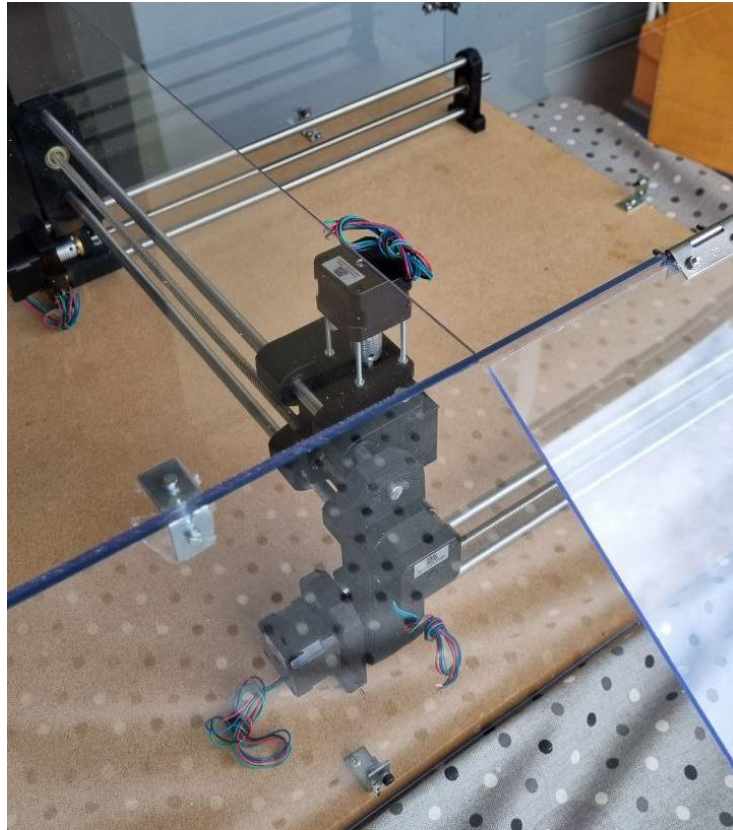
Il·lustració 1712: Motor de l'eix Z

A continuació, instal·lem l'altre costat de l'aparell seguint els mateixos passos que l'anterior i instal·larem l'última claveguera.



Il·lustració 18: Màquina amb tots els components mecànics muntats

Per acabar instal·larem la càpsula utilitzant les esquadres i la tapa, collant-li les frontisses.



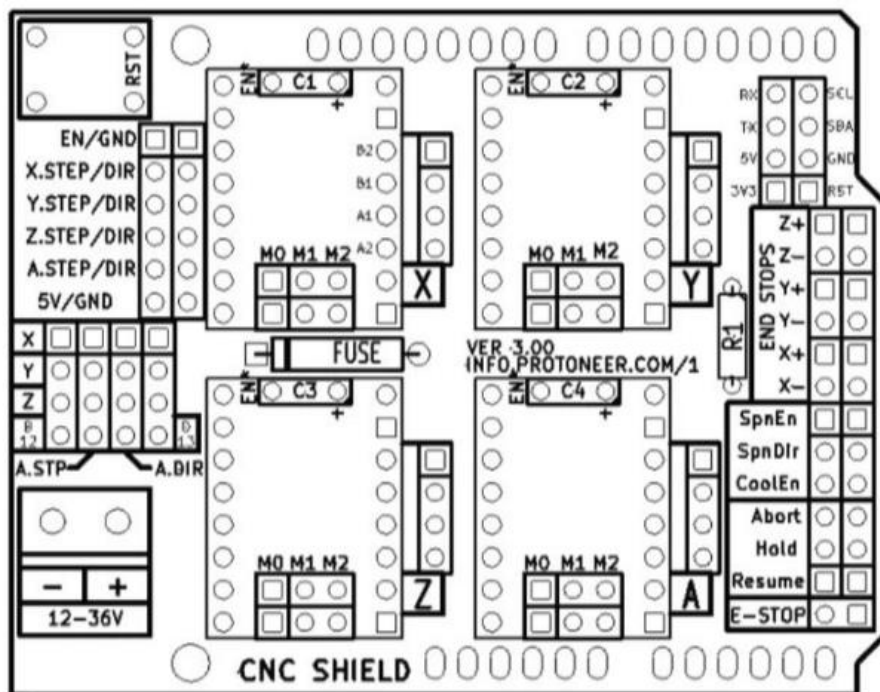
Il·lustració 13: Aparell dins la seva cabina

5.4 Connexionat

Per a poder controlar els motors pas a pas utilitzant els drivers de control utilitzarem una placa CNC Shield. El que s'aconsegueix utilitzant la placa CNC shield és simplificar enormement el cablejat entre la placa d'Arduino i els motors pas a pas.

A l'hora d'alimentar la CNC shield s'ha de tenir en compte el seu rang de tensió que és entre 12 i 36V. En aquest cas s'alimentarà a 12V utilitzant una font externa.

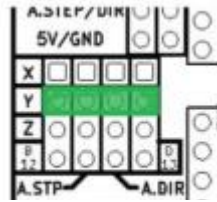
Per a conèixer detalladament els pins de connexió que s'utilitzaran, es mostra a continuació un esquema de connexions de la Shield:



Il·lustració 2014: Esquema de la CNC Shield

Com es pot apreciar en la il·lustració 14, la Shield només pot integrar 3 eixos (X, Y, Z). Hi ha la possibilitat de clonar un quart eix (A). Aquest té com a missió principal atorgar major força motriu a un dels 3 eixos. En el nostre cas, es clonarà l'eix Y.

Per a configurar el clon de l'eix Y s'han de connectar uns punts de derivació en la part de la Shield on es poden apreciar els pins de configuració per a: A.STP (Passos) i A.DIR (Direcció o sentit de gir del motor). Per a aconseguir aquesta configuració, hem de connectar en la nostra placa la següent combinació de pins:



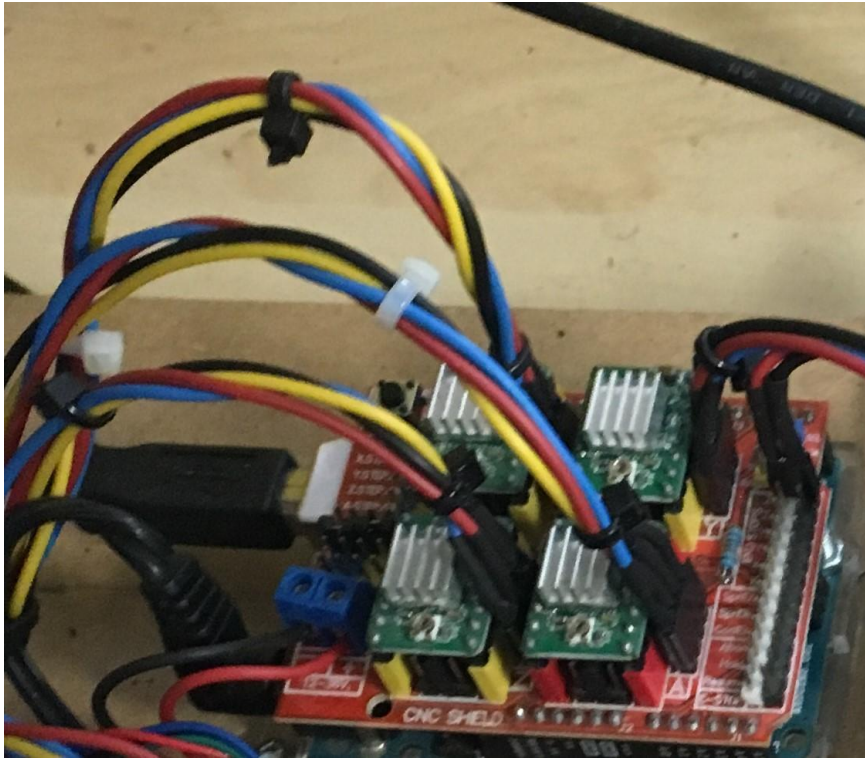
Il·lustració 21: Esquema dels punts per clonar eix

També s'han de puntejar els pins M0 i M1 de cada driver per aconseguir el 1/8 de pas explicat en apartats anteriors.

Per a connectar els motors als drivers primer s'ha de calcular el valor de referència que han de tenir els drivers. Aquest depenen de la intensitat de fase de cada motor i es treu de la següent equació.

$$V_{REF} = I_{MAX} \cdot (8 \cdot R_S)$$

On R_S és la resistència de sensat del driver, en el cas d'aquest driver té un valor de $0,1\Omega$, i I_{max} la intensitat de les bobines del motor. El resultat d'aquesta operació dona una V_{ref} de $1,064V$. Per seguretat s'agafa un valor del 75% d'aquest valor per protegir els drivers i motors de possibles sobreimpulsos, per tant el valor que s'ha d'aconseguir de V_{ref} es de $0,798V$. Després de calcular V_{ref} mirem s'ha de orientar el potenciòmetre del driver per que a la seva sortida dongui el valor esmentat. Per moure rotar el potenciòmetre farem servir d'un tornavís d'estrella petit i ens ajudarem d'un voltímetre per medir la tensió volguda. Un cop fet això es connecten les diferents bobines dels motors seguint el codi del fabricant i connectant-los als pins corresponents de la CNC Shield.



Il·lustració 22: Cablejat CNC Shield

Per acabar es connecten els sensors de final de carrera, els indicadors leds, els diversos botons que actuen a la maquina als pins lliures de la placa d'Arduino per poder-los controlar més endavant. En el cas del làser es connectara a un pin amb sortida SPWM.

5.5 Programa de visió artificial

Com s'ha comentat en l'abast del projecte, les peces que intentarem replicar són objectes simples els quals no tindran cap línia que no sigui rectilínia. Això permet facilitar en gran mesura el programa de visió ja que l'únic que caldria per replicar la imatge seran les cantonades de la figura a replicar per després poder unir-les entre elles en l'ordre correcte. Per poder fer aquesta detecció utilitzarem l'algoritme de detecció de Harris.

5.5.1 Detecció de cantonades Harris

Aquest operador de detecció de cantonades fou introduït per primera vegada per Chris Harris i Mike Stephens en 1988 després de la millora del detector de cantonades de Moravec. El seu funcionament és el següent:

A la imatge se li fa un anàlisi de components principals (PCA en anglès). Aquest anàlisi permet veure els diferents gradients de variació de cada píxel amb el dels seus veïns i representa aquestes variacions en un gràfic 2D. L'ACP construeix una transformació lineal que escull un nou sistema de coordenades pel conjunt original de dades en el qual, la variància de major mida del conjunt de dades és capturada en el primer eix, primera component principal. La segona variància més gran és el segon eix, segona component principal, i així successivament. Per construir aquesta transformació lineal s'ha de construir, primer, la matriu de covariància o matriu de coeficients de correlació. Degut a la simetria d'aquesta matriu existeix una base completa de vectors propis que denominarem com λ_x . En el nostre cas la matriu tindrà 2 vectors propis que seran els valors que determinaran si la "finestra" escollida és o no un vèrtex. Hi ha 3 possibles casos:

- LAMBDA1 i LAMBDA2 són valors petits al voltant de 0 per tant ens trobem amb un finestra on no hi ha variacions de la intensitat i per tant és una regió homogènia.
- LAMBDA1 o LAMBDA2 tenen un valor elevat. Això determina que la regió només té variabilitat en un dels eixos i per tant ens trobem amb un marge de la figura.

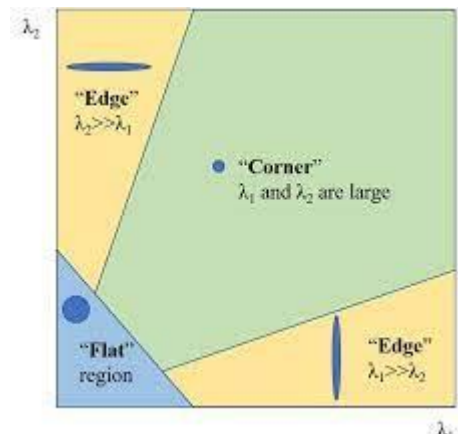
LAMBDA1 i LAMBDA2 tenen valors elevats. Com que tenim una gran variabilitat en els dos eixos podem determinar que es tracta d'una cantonada.

Aquest algoritme s'implementa amb la llibreria OpenCV per mitjà de la funció "goodFeaturesToTrack". Aquesta necessita passar-li els següents paràmetres: la imatge la qual es vol aconseguir les cantonades, el vector com a variable resultant de l'operació, el màxim número de cantonades, el valor mínim de qualitat en tant per un per determinar si un punt es pot considerar com a cantonada, la distància mínima entre cantonades, una màscara opcional en la qual determina en quina zona de la imatge es busquen les cantonades, el tamany dels blocs que s'utilitzen per fer comparacions entre ells per els càlculs pel detector Harris, un bolea que determina si s'utilitza o no el detector de cantonades Harris i la constant utilitzada pel detector.

El valor mínim de qualitat funciona de la següent forma: l'algoritme busca la cantonada amb el valor mínim propi major i llavors multiplica aquest valor pel valor de qualitat.

El valor resultant és el llindar de valor mínim propi el qual un punt ha de superar per determinar si un punt és una cantonada.

Un cop obtinguts els punts de les cantonades estan ordenats de major a menor respecte al valor mínim propi de cadascú. Per poder reproduir la imatge de forma correcta s'han de reorganitzar els punts.



Il·lustració 23: Detector de cantonades Harris

5.5.2 Algorisme d'ordenació dels punts d'interès

Com s'ha explicat anteriorment és molt important poder tenir un ordre lògic en els punts per saber quin moviment ha de fer la màquina en tot moment. Per poder obtenir una solució simple a aquest problema s'afegeix la hipòtesi de que A l'haver fet la hipòtesi que utilitzarem les formes que volem utilitzar són formes geomètriques simples utilitzarem un algoritme simple el qual buscarà ordenar els punts segons el seu angle. Per això el programa comença trobant el punt central de l'objecte a replicar. S'aconsegueix el centre fent un sumatori de les components X i Y per separat i dividir els resultats entre el nombre de punts obtinguts. Un cop aconseguit, normalitzarem cada punt restant les coordenades del centre a cada un per aconseguir els punts de les cantonades relatius al centre. Després només fa falta trobar l'angle de cada punt amb la funció "atan2" la qual fa la inversa de la tangent fent servir els valors de x i y. La funció dona de rang de - a, però per poder obtenir una ordenació lògica hem d'aconseguir que el rang passi a ser de 0 a 2. Per això sumarem cada angle 2 per obtenir el nostre objectiu. Ara ja es poden ordenar els angles de major a menor i així aconseguirem un ordre en el sentit de les agulles del rellotge del vector d'angles, però el que es vol ordenar són els punts, no només els angles. Per aconseguir ajuntar el resultat d'angles als punts s'adjunten al valor d'angles les coordenades que representen aquell angle usant 3 dígits per l'angle i 5 dígits per cada coordenada. Si el valor d'alguna coordenada fos negatiu, ja que a l'estar normalitzat amb el centre hi haurà valors negatius, s'intercanviarà el signe "-" per un 0 a l'inici de cada coordenada. De la mateixa forma s'eliminarà el símbol de la coma. Un cop realitzat aquests canvis l'únic que ens cal és ordenar de major a menor els angles i, un cop reorganitzats, recuperar els punts units als angles tornant tant els valors negatius com les comes a on corresponguin i sumar-li el centre a cada punt per restaurar els valors inicials però en aquest cas ja ordenats.

Un cop fet el procés d'obtenció dels punts i d'ordenar-los en l'ordre establert es pot començar a determinar el moviment del robot.

5.6 Programa de control del moviment

El punt inicial o "home" del robot serà la cantonada esquerra de la nostra rèplica. Aquest punt serà on el robot torni a l'acabar el procés o si hi hagués una parada d'emergència. Un cop determinat el punt inicial es calcula el pendent de l'equació de la recta entre el punt inicial i el primer punt de la figura i la distància entre ells. El pendent s'aconsegueix amb la següent fórmula:

El pendent d'una recta és la mesura de la inclinació de la recta. Més en concret és la longitud que la recta recorre en direcció vertical quan recorre una unitat de longitud en direcció horitzontal.

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Per aquesta raó en comptes d'agafar el valor real del pendent s'utilitza la fracció simplificada per obtenir els valors enters de les coordenades X i de Y sense tenir la necessitat d'utilitzar un valor amb diverses xifres significatives.

Un cop obtinguda la fracció s'emmagatzema els valors del numerador i denominador que seran els polsos que activarem els motors. També es buscarà el quadrant en què s'haurà de moure la màquina comparant les coordenades del següent punt amb l'actual. Un cop ocupat el tema per saber la direcció del moviment buscarem quina distància s'ha de recórrer. Per aconseguir aquesta distància calcularem el mòdul del vector resultant. Un cop aconseguit el mòdul dividirem aquest valor pel pas de rosca de l'eix per saber quantes voltes ha de fer l'eix per fer aquesta distància. Després es multiplicarà el valor per 8, ja que com s'ha explicat en l'apartat del driver de control dels motors el nostre driver l'haurà configurat perquè cada pols en què l'excitem el motor farà 1/8 de volta. Per aquesta mateixa raó podem extreure que l'error mínim al qual podem aspirar és igual al pas de rosca de l'eix multiplicat pel gir que fa l'eix en un pols. En aquest cas l'error serà de ± 0.005 mm.

Un cop calculada la distància farem el mateix càlcul del mòdul de la fracció del pendent per esbrinar la distància que ens movem per cada cicle determinat per la fracció i dividirem la distància total per la distància de cicle per calcular el nombre de cicles que ens haurà de moure. Si el nombre no és enter a l'últim cicle multiplicarem els dos números de la fracció pels 2 decimals més significatius i ens quedarem amb el valor arrodonit a l'enter més pròxim per simplificar els càlculs. Un cop extreta la informació del moviment s'ha de poder comunicar d'una forma simple cap a la placa d'Arduino que s'encarregarà de fer el control actiu dels motors.

5.7 Protocol de comunicació

Un cop executat el programa tenim diversa informació emmagatzemada que hem de passar a la placa d'Arduino per poder fer el moviment requerit en tot moment. Aquesta informació es tracta de: el nombre de polsos per eix de coordenades que s'ha de fer per moure's d'un punt a un altre, el nombre de cicles per arribar a aquest punt i la direcció determinada pel que hem nomenat quadrant i el signe del pendent. Aquestes 4 dades es passaran a l'Arduino en connectar-lo a través del port USB. Per poder comunicar-se el nostre programa amb l'Arduino implementarem una comunicació via port sèrie i d'aquesta manera podrem tant enviar com rebre informació. Per poder fer aquesta connexió crearem una nova variable de tipus "SerialPort" i en aquesta li passarem el nom del port en què s'hagi configurat l'Arduino, el baudrate en el qual es produirà la comunicació, si té paritat, el nombre de bits per trama, el nombre de bits de parada i el tipus de protocol de comunicació que es farà servir. Un cop establert totes aquestes variables s'envia la consigna d'obrir el port. A l'obrir el port el programa enviarà la paraula "START" perquè el programa d'Arduino estigui preparat per començar. Després de rebre aquesta consigna el programa esperarà que es compleixi les condicions dels sensors que marquen que la tapa està tancada per respondre al programa de visió amb un "CLOSED" i donar llum verda a què se li comenci a passar les consignes de moviment.

Per reduir la complexitat de la programació de l'Arduino, ja que està més limitada que la que programem en ordinador, el programa de visió enviarà només una trama d'informació al moment i llavors esperarà fins que es confirmi que la màquina ha arribat fins al punt requerit per poder enviar la següent trama. Si en algun moment s'aprestés el botó d'emergència, l'Arduino enviaria les variables per saber en quin moment del cicle s'ha quedat aturat per després calcular el moviment cap a l'"home" i, un cop donat el botó de "rearmament", s'enviaria les dades i es donaria a terme el moviment requerit per restablir la posició inicial. Per comunicar aquesta pausa el robot enviarà la paraula "STOP" tot seguit del valor de x i y que li faltés per acabar el cicle juntament amb el valor resultant de cicles que ha dut a terme. Un cop acabat el cicle enviarà la paraula "OK" per confirmar que ha arribat al punt indicat.

Un cop explicat el programa de visió es passarà a explicar el programa que de l'Arduino.

5.8 Programa de l'Arduino

El programa de l'arduino no només és l'encarregat de fer el moviment del robot, sinó que a més és el que es fa càrrec dels diferents actuadors que hi ha a la càpsula.

El programa una vegada iniciat i que s'hagi dut a terme l'inici de la comunicació s'espera a la confirmació que la tapa està en posició tancada. Aquesta confirmació és definida pels sensors de final de carrera de la tapa. Els sensors funcionen com ha interruptors normalment oberts, que quan la tapa està baixada es tanca el circuit i arriba alimentació als pins X i Y de l'Arduino. En arribar aquesta confirmació s'activa un led verd per mostrar que la màquina està encesa i preparada per funcionar.

Com s'ha explicat durant la part del programa de visió, s'ha buscat simplificar al màxim el programa d'Arduino corresponent al moviment de motors. Quan arriba la informació del moviment es crea 2 bucles, un amb els passos de coordenades pertinent i l'altre un per anar repetint aquesta primera iteració les vegades necessàries per arribar a on toca. A l'acabar aquest segon bucle s'envia la verificació pertinent per la comunicació del port sèrie.

Si en algun moment hi hagués algun problema es pot prémer el commutador per parar la màquina de forma immediata. Quan es pressiona el commutador, arriba l'alimentació en el relé que aporta l'alimentació als motors i al làser. Aquest a l'estar connectat com a normalment tancat quan li arriba l'alimentació nominal a la bobina del relé obre el circuit impossibilitant el moviment de la màquina. Mentre es manté pressionat l'interruptor s'il·lumina el llum i és desbloquejar la tapa per fer les reparacions necessàries. També mentre està parat primer es comunica la parada al programa de visió i després s'envia la informació per saber on s'ha quedat la màquina.

Un cop obert l'interruptor, el programa espera que s'apreti el polsador de "rearmament" perquè s'envii la consigna "OK" i el programa de visió envii les indicacions per tornar a la posició inicial.

6. Pressupost

El cost total del projecte han estat 490,46 €. La part electrònica del projecte, és a dir motors, placa Arduino o pilots leds per posar un exemple té un cost de 197,26 €. La part mecànica té un import final de 293,20 €. Els components que tenen més pes en aquest valor són, d'una banda, les plaques de metacrilat per conformar la cabina amb un preu de 121,80 € i els motors pas a pas amb un import de 101,48 € en total. Es pot trobar el pressupost desglossat en el document "Pressupost".

Cal destacar que en aquest projecte no s'ha comptat amb el cost d'impressió de les peces en 3D, ja que es tenien els recursos necessaris per a poder-les fabricar sense cost aparent. Si s'hagués hagut de demanar la impressió d'aquestes a una empresa, el pressupost hauria augmentat entre un 30 i un 40% arribant pràcticament als 700 €.

7. CONCLUSIONS

7.1 Resultats

Les proves han estat realitzades mitjançant 2 peces diferents. La primera s'ha fet reproduint una peça quadrada. Aquesta forma simple permet veure el moviment per separat dels eixos, ja que la línia resultant entre cada aresta té o pendent 0 quan es mou en la coordenada X o pendent infinit en valor absolut quan el moviment es produeix en l'eix de la Y. La segona figura ha estat una figura triangular amb els seus costats irregulars. Amb aquesta figura s'hi afegeix la dificultat de fer rectes on la màquina es mou en els dos eixos. En moure's en els dos eixos el que pot passar és que hi hagi un problema de sincronisme entre els dos moviments i que no es produeixi una línia recta. S'ha buscat un triangle amb tots costats diferents per comprovar que el sincronisme funciona bé encara que hi hagi una variació més gran en el moviment que s'ha de fer en els dos eixos en tot

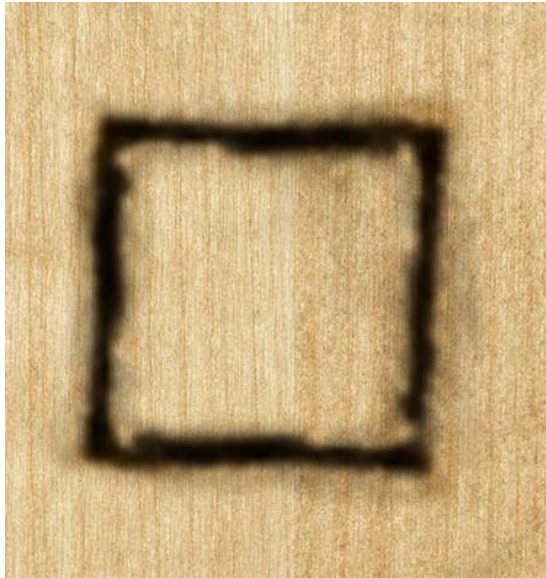


moment.

Il·lustració 24: Peces a replicar

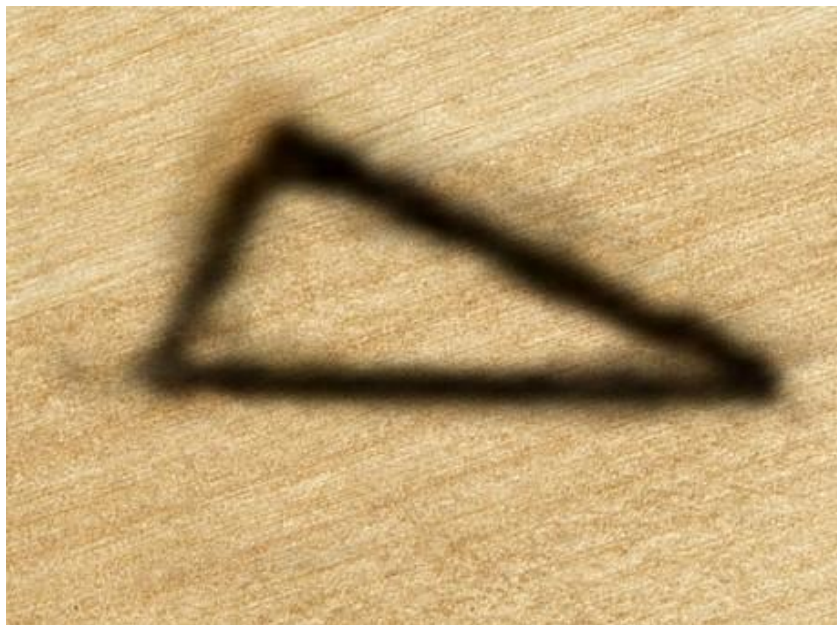
Les proves s'han fet en dos trossos idèntics de fullola de fusta amb la mateixa dimensió que la zona útil de la màquina per a ser gravats pel làser.

Els resultats han estat els següents:



Il·lustració 2515: Resultat primera prova

Com es pot observar la primera prova ha produït una representació bona de l'objecte donat. El contorn és més o menys clar i té petites diferències mínimes amb la peça que volíem replicar, potser degut al material en el que s'ha intentat gravar.



Il·lustració 16: Resultat segona prova

En la segona prova es pot posar de manifest que la figura resultant no ha sortit amb el nivell de nitidesa que la primera. La peça ha sortit clarament menys nítida augmentant l'error en els costats on s'havia de moure en diagonal. Això pot ser degut a una falta de control de la sincronia del moviment en els dos eixos. També pot haver afectat la necessitat

de més rigidesa per part del sistema perquè, quan el moviment no és del tot regular continuï fent la seva funció sense que interfereixi en el resultat final. Una possible solució seria puntejar tots els pins del driver per fer que cada pols signifiqui $1/16$ de pas en comptes de la configuració actual d' $1/8$. Aquest fet implicaria un augment de la precisió tot i que aniria amb detriment del temps del procés.

7.2 Millores

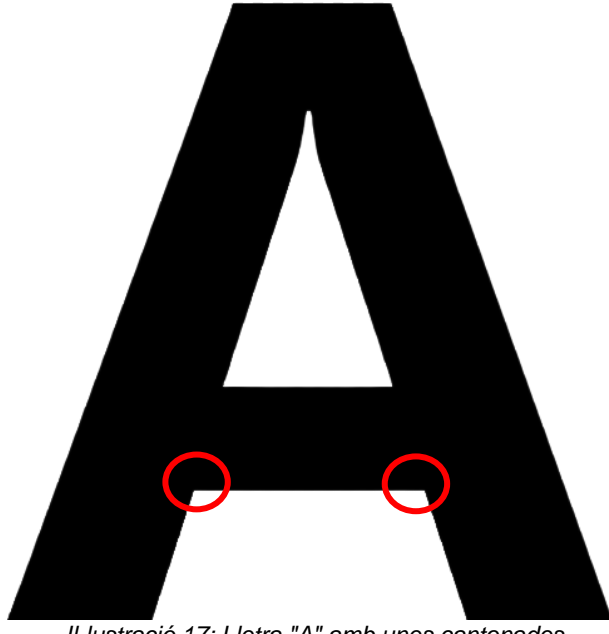
Un cop realitzat les conclusions de les proves es comentarà les possibles millores de la màquina.

En dissenyar totes les parts de la màquina de forma autònoma totes tenen una certa simplicitat que fa que hi hagi molts aspectes a millorar

Per la part de hardware una de les possibles millores seria la de canviar els materials de construcció de la màquina. Si es canviés la fusta i el metacrilat de la càpsula i el plàstic de les peces en 3D per un material ferros la màquina guanyaria robustesa i durabilitat, però en contraposició augmentaria significativament el cost de construcció de la màquina. Una altra de les millores possibles podria ser el de preparar la màquina per poder fer altres funcions. Amb el disseny del capçal corresponent i amb la inclusió de les eines adequades seria relativament senzill optimitzar la màquina perquè fes treballs de fresat o de soldació per posar un exemple.

Un altre canvi a millor seria el d'intercanviar els motors pas a pas per uns motors de corrent continu convencionals. Els motors pas a pas són motors que ofereixen una simplicitat evident en el seu control, però tenen limitacions en el seu moviment. A l'utilitzar motors de CC i fent servir un algoritme de control als motors s'aconseguiria augmentar les capacitats de la màquina.

La part del software és de les que més simple és. Una de les coses que s'ha volgut simplificar més és en l'obtenció dels punts característics per reproduir la forma i la manera en què s'ordenen d'aquests. L'algoritme d'ordenació és funcional dins la hipòtesi de reproduir figures simples, però té certes limitacions quant a fer figures amb una certa complexitat. Per exemple si la figura no fos regular i l'ordre de les cantonades no es pogués obtenir només ordenant els angles la replica, resultant no seria representativa. A més d'aquest problema s'hi afegix el fet que no es pot reproduir una imatge que no sigui uniforme en el seu interior, és a dir, que tingui més d'un contorn. Per exemple agafant la següent figura els punts marcats deixen de seguir la lògica de què totes les cantonades segueixen un ordre horari entre elles respecte als seus angles i a més té dos contorns clarament diferenciats.



Il·lustració 17: Lletra "A" amb unes cantonades assenyalades

Una possible solució de la primera dificultat seria que a part de trobar les cantonades també es busqués el punt intermedi de cada costat i així crear una equació de la recta entre una cantonada i el punt mig per veure la recta resultant a quina cantonada s'aproxima. Pel que fa a la segona complexitat el programa hauria de diferenciar els contorns i reproduir el resultat per a cada contorn per separat. D'altra banda, una de les hipòtesis inicials del projecte determina que les peces a reproduir han de ser rectilínies, per tant, la forma que s'ha triat per aconseguir els punts per reproduir la imatge fa que no es puguin reproduir formes que tinguin arcs, només aquest tipus de peces. Això es podria millorar si en comptes de buscar les cantonades es fes un mostratge del contorn de la figura amb l'interval prou petit perquè en unir els punts en una recta al final el que quedi representat sigui l'arc corresponent. D'aquesta forma la dificultat augmentaria no només pel fet d'augmentar la dificultat d'obtenció dels punts i l'increment del número d'aquests sinó que a més s'haurien de fer diversos càlculs o experiments per determinar l'espai mínim entre punts per a realitzar una bona representació, però que no faci que el nombre de punts sigui tan elevat que superi els recursos del programa.

Un altre afegiment sense canvis en l'estructura de la màquina seria afegir la funció de gravat. Per fer aquesta funció primer es passaria la imatge, la

qual es volgués reproduir en un material determinat, a una imatge d'escala de grisos. Un cop fet això es detectaria els píxels que no estiguessin en blanc i es guardaria el valor de la seva intensitat de color. Un cop fet aquest pas la màquina aniria fent moviments de manera que anés passant fila per fila activant i regulant el làser quan passes per píxels que tinguin assignats valors de colors diferents de 0.

Implementar una interfície també suposaria un salt respecte a tal com està el projecte actualment. Per les complicacions ocasionades durant la realització de la màquina no hi ha hagut temps per implementar alguna encara que fos simple. La idea que tenia per fer-la era una simple que només mostres per pantalla l'estat de la màquina. Amb una mica de comunicació extra es podria realitzar una interfície que pogués actuar en la màquina i, inclús, fer una representació de com està anant el procés en temps real en forma de dibuix. Aquesta interfície podria arribar a ser mostrada en una petita pantalla fixada a la paret de la càpsula.

Per acabar amb les possibles millores la que tindria un impacte major tant en l'increment de dificultat com a l'augment d'opcions seria el d'afegir un tercer eix a la màquina. Això permetria que el tall es fes en peces 3D, molt més complexes que les 2D actuals. Aquest canvi afectaria tant a la part de hardware com de software. En la part de hardware a part dels canvis evidents per afegir el tercer eix s'hauria d'augmentar el tamany de la càpsula i veure com subjectar la peça per poder-la treballar de la forma més eficient possible. També la part del mecanitzat hauria de canviar a fer fresat en comptes de tall per làser, ja que al ser peces més grans el làser no seria tan eficient.

Per part de software, la part de control del moviment elevaria la dificultat actual, ja que s'hauria d'afegir la variable de l'angle en el qual l'eina ha d'actuar. També deixaria de ser viable el tema de reproduir el contorn d'una imatge donada i s'hauria de passar la peça a través d'un programa de CAD i traduir-la mitjançant un programa extern per aconseguir un llistat dels moviments que ha de fer la màquina semblant al codi "GCode".

8. BIBLIOGRAFIA

[1] Anònim.(2020, Octubre 1).Wikipedia [Online].Disponible:
https://es.wikipedia.org/wiki/John_T._Parsons.

[2] F. Shawn. (2021, Octubre 15). Thomasnet [Online]. Disponible:
<https://www.thomasnet.com/articles/custom-manufacturing-fabricating/what-does-cnc-stand-for/>

[3] S.Gray. (2021, Octubre 15). Hubs [Online]. Disponible:
<https://www.hubs.com/guides/cnc-machining/>

[4] Anònim.(2022, Març 3).Wikipedia [Online].Disponible:
https://en.wikipedia.org/wiki/Laser_cutting

[5] Anònim.(2021, Abril 28). Forum Arduino [Online]. Disponible:
<https://forum.arduino.cc/t/use-cnc-shield-without-grbl/854919>

[6] Sánchez, Javier; Monzón, Nelson; Salgado, Agustín.(2018, Octubre 3). “*An Analysis and Implementation of the Harris Corner Detector*” [Online]. Disponible:
http://www.ipol.im/pub/art/2018/229/article_lr.pdf

[7] CRUZ TERUEL, FRANCISCO.(2004, Desembre 13). “*Control Numérico y Programación*”. Editorial Marcombo. ISBN: 978-84-267-1359-9.

[8] Anònim. (2013) Instructables [Online]. Disponible:
<https://www.instructables.com/Building-a-CNC-router/>