THE UNIVERSITY OF DANANG
UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

PBL 4: DESIGN OF ELECTRONIC SYSTEMS

Group 20.04C

PROJECT: LASER ENGAVING MACHINE

Lecturers:	**Ph.D TRAN DINH SON**
Student:	**VO TAN PHU**
	PHAN HUU THANG
Student ID Number:	**101200281**
	101200242
Class:	**20CDT2 & 20CDT1**

Danang, January 2024

DA NANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

THE SOCIALIST REPUBLIC OF VIETNAM
Independent – Freedom - Happiness

MISSION OF PROJECT PBL4

Num	Name	ID	Class	Industry
1	Vo Tan Phu	101200281	20CDT2	Mechatronic Engineering
2	Phan Huu Thang	101200242	20CDT1	Mechatronic Engineering

1. Lecturer: PhD. Tran Dinh Son, PhD. Pham Anh Duc
Name of project: Laser Engaving Machine 2 Axis

2.The subject is subject to: ☐ Having signed an agreement on intellectual property for the results of implementation

3. Content of the explanations and caculations:

a) General part:

Num	Name	Content
1 2	Vo Tan Phu Phan Huu Thang	- Get an overview about Laser Engaving Machine - Hardware and software design caculation - Prepare presentation, slides and drawings as required

b) Private part:

Num	Name	Content
1	Vo Tan Phu	- Calculation and design of electronic and control parts - Design HMI interface to monitor system operation
2	Phan Huu Thang	- Calculation of mechanical design - System control programming

4. Drawing, graphs (specify types and sizes of drawings):

a) General part:

Num	Name	Content
1 2	Vo Tan Phu Phan Huu Thang	- Make the overall drawing of the machine (1 – A0)

b) Private part:

Num	Name	Content
1	Vo Tan Phu	- Make an algorithm flowchart drawing
2	Phan Huu Thang	- Make a drawing of an electrical circuit diagram

5. **Project assignment date:** 21/8/2024
6. **Project completion date:** 9/12/2024

DA NANG UNIVERSITY OF SCIENCE AND TECHNOLOGY	THE SOCIALIST REPUBLIC OF VIETNAM
	Independent – Freedom - Happiness

PBL4 PROJECT IMPLEMENTATION SCHEDULE

Time	Content
Week 1 (21/8 – 27/8)	Meet with teacher to discuss the topic
Week 2 (28/8 – 3/9)	Find out about the project
Week 3 (4/9 – 10/9)	Learn about different types of sensors and get ideas for projects
Week 4 (11/9 – 17/9)	Project report to teacher
Week 5 (18/9 – 24/9)	Draw schematic diagrams, 3D printed circuits and models for the system.
Week 6 (25/9 – 1/10)	Report the overall drawing of the model to the drawing A0. Report 3D printed circuit diagram, export layout, report product cost.
Week 7 – 12 (2/10 – 12/1)	Project programming
Week 12–16(13/11–19/11)	Programming interfaces and project reports.
Week 15 (27/11 – 3/12)	Finalize the project, overall report.
Week 16 (4/12 – 10/12)	Complete project

CONTENT

INTRODUCTION

In the current era of industrialization and modernization, the Mechanical Engineering industry has become a cutting-edge sector with significant importance for various industries, asserting its prominent position.

As mechanical engineering students at the University of Technology, after four years of studying and training under the dedicated guidance of our professors, we have gained valuable knowledge. For our assigned project on the Design of Electronic Systems, we have decided to collaborate on the project "Laser Engraving Machine Design." Given the increasing demand for tourism and engraving services on daily items, we find it essential to undertake this Project

Throughout the project, due to time constraints and limited specialized knowledge, this is also our first attempt at a technical electronic design, and we anticipate some inevitable shortcomings and errors..

We sincerely look forward to constructive feedback from our esteemed professors to successfully complete this design project.

I express my sincere gratitude to all the professors in the Mechanical Engineering department, especially to Professor Tran Dinh Son, for their guidance and support throughout this project.

Sincerely
VO TAN PHU

CHAPTER 1 INTRODUCTION TO CNC AND LASER ENGRAVING MACHINES

1.1 Setting the problem

- Laser technology is becoming increasingly crucial in delivering high-quality products for the supporting industries. In Vietnam, industrial laser equipment has been present in the market for many years, mostly supplied by commercial enterprises. CNC machine enable the processing of highly accurate and complex products that traditional can run automatically until completion, thus freeing up labor for other tasks.

Figure 1.1 Laser engraving technology

- In line with this trend, with the aim of creating a precision tool capable of drawing, engraving on wood, plastic, paper, leather, fabric, serving arts and crafts, souvenirs, and tabletop pictures automatically, our team has undertaken this project. Additionally, the product is manufactured with an aesthetically pleasing design, low production costs, and maintenance compared to similar foreign products, making it affordable for users. The product is a specific application of a high – precision automation system in daily life and production, providing higher efficiency compared to traditional engraving methods. Equipped with user-friendly visual support software, the product can be easily by anyone after a few hours support software, the product can be easily used by anyone after a few hours of learning, making it potentially widely applicable and popular.

The situation within the country

- while CNC laser technology has been present worldwide for a long time, it is only in recent years that Vietnam has begun to focus on in-depth research into this technology for industrial applications.
- Currently, in Vietnam, there are also some research topics serving teaching related to CNC milling machines, CNC plasma, CNC laser…

+ Some laser engraving machine currently available on the market include:

Figure 1.2 Some types of machine Laser

1.2 The purpose of researching this topic

- The aim of this new research, design, and manufacturing direction is to develop and enhance a CNC Laser machine with a simple and flexible structure and control system, ensuring high efficiency in product quality, and particularly achieving a low- cost production.
- Additionally, CNC machines are a rapidly advancing field worldwide. Therefore, researching, designing, and manufacturing a CNC Laser Engraving Marching using this Arduino control system provides us with exposure to a vast amount of knowledge in electronics, computer science, and mechanics that we may not have learned in school. This experience helps us feel more confident when dealing with read-world applications.

- The research and manufacturing process will encounter various challenges and situations that require individuals to address practical issues. This experience not only contributes to the successful completion of the project but also serves as a valuable opportunity for personal growth and the accumulation of experience for future endeavors.

The objectives of the project :

+ Research and explore topics related to CNC laser both domestically and internationally.
+ Design and manufacture the mechanical components of the CNC laser machine.

+ Research, study, and assemble the electrical – control components of the CNC Laser machine.

1.3 CNC Basic Concepts:

1.3.1 CNC Machine Concept:

- CNC (Computer Numerical Control) is a type of automated NC machine assisted by a computer, where the automatic components are programmed to operate in a sequence of events at a predetermined speed to produce a workpiece with the required shape and dimensions

1.5.2 CNC Machine Axes:

- In order to control the tool movement along geometric paths on the surface of the workpiece, there must be a relationship between the tool and the workpiece. This relationship can be established by placing the tool and the workpiece in a coordinate system. The Decac coordinate system in commonly used in CNC machine.
- During this process, the space is limited by the three dimensions of the coordinate system attached to the machine that the machine control system can recognize, known as the machining envelope,

1.5.3 Principles of Establishing Coordinate System on CNC Machines:

- To determine the coordinate axes, we rely on the right-hand rule, which involves the middle finger, index finger, and thumb of the right hand. The thumb indicates the direction of the x-axis, the index finger points along the Y-axis, and the middle finger specifies the Z-axis.

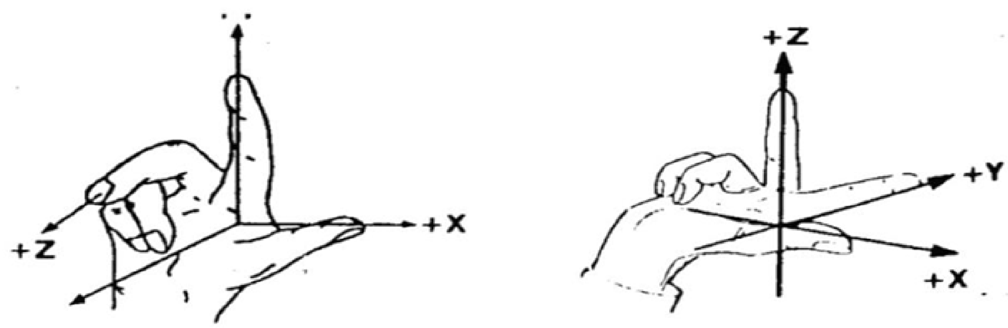


Figure 1.3 Right-Hand Rule.

- To determine the coordinate axes of the NC machine through this principle, first, we imagine that the middle finger is aligned with the main spindle of the machine, representing the Z-axis, with the positive direction of the axis pointing from the inside to the outside. Accordingly, the thumb and index finger will indicate the orientation and direction of the X-axis and Y-axis.
- The rotary axis is determined by the straight axes on which the cutting tool rotates. A is the rotary axis on the X-axis, B is the rotary axis on the Y- axis, and C is the rotary axis on the Z-axis. When looking in the positive direction of the main axes, the clockwise direction is considered the positive direction of the rotary axes.

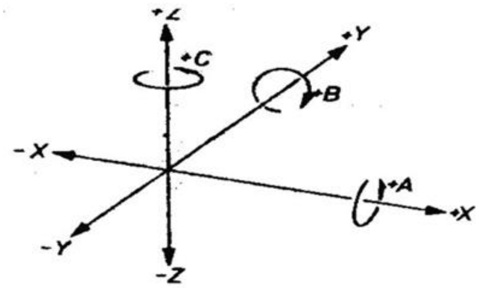
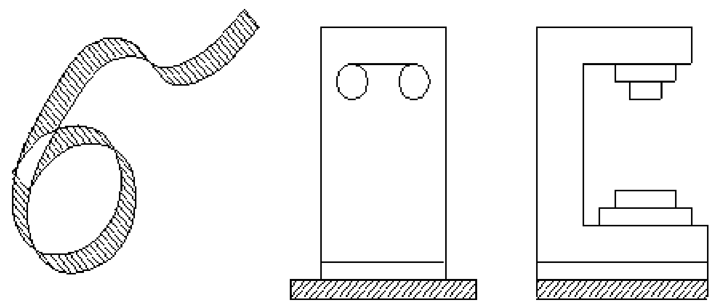


Figure 1.4 Three rotary axes A, B, C

1.5.4 CNC Machine basic components:



1.Program 2. Control System 3. Machine tool

Figure 1.5 Basic components of the System

a) Control program

- This is a set of commands that the machine must execute. These commands are encoded in numerical and symbolic form that the control device can recognize. The control program can be stored on punched cards, punched tape, or magnetic tape.

b) Control System

- The second component of the digital control system. It comprises electronic circuits and hardware capable of reading and interpreting the control program and transmitting it to the machine tool. The basic components of the control unit include

Data Storage Unit

+ Data Distribution Unit
+ Feedback Unit
+ Sequential Control Unit to coordinate the operation of the elements. It is important to note that nearly all modern CNC machines are equipped with a control unit known as a Microcomputer. Hence, they are referred to as CNC machines

c) Machine tool

- The machine tool comprises the machine table and spindle, as well as the motors and necessary controls for the machine to operate. It also includes cutting tools, fixtures, and other auxiliary devices required for machining.
- CNC machines vary widely, ranging from simple hole drilling and milling machines to sophisticated machining centers.

1.5.5 Interpolation Algorithm on CNC Machines

a) Straight Line Interpolation Method

- Concept: In computer numerical control (CNC) machines, the toolpaths on the workpiece are formed by coordinating movements along multiple axes. The nozzle is moved from the starting point to the endpoint in a sequence of straight line segments.
  + Implementation: straight line interpolation along 2 or 3 axes is very common and involves the following required parameters.
  + Starting point coordinates, ending point coordinates
  + Speed of movement on each axis.
- Capability: In theory, straight line interpolation can program any curved trajectory, but the amount of data to be processed is very large. Using circular, parabolic, or helical interpolation significantly reduces the amount of programmed data.

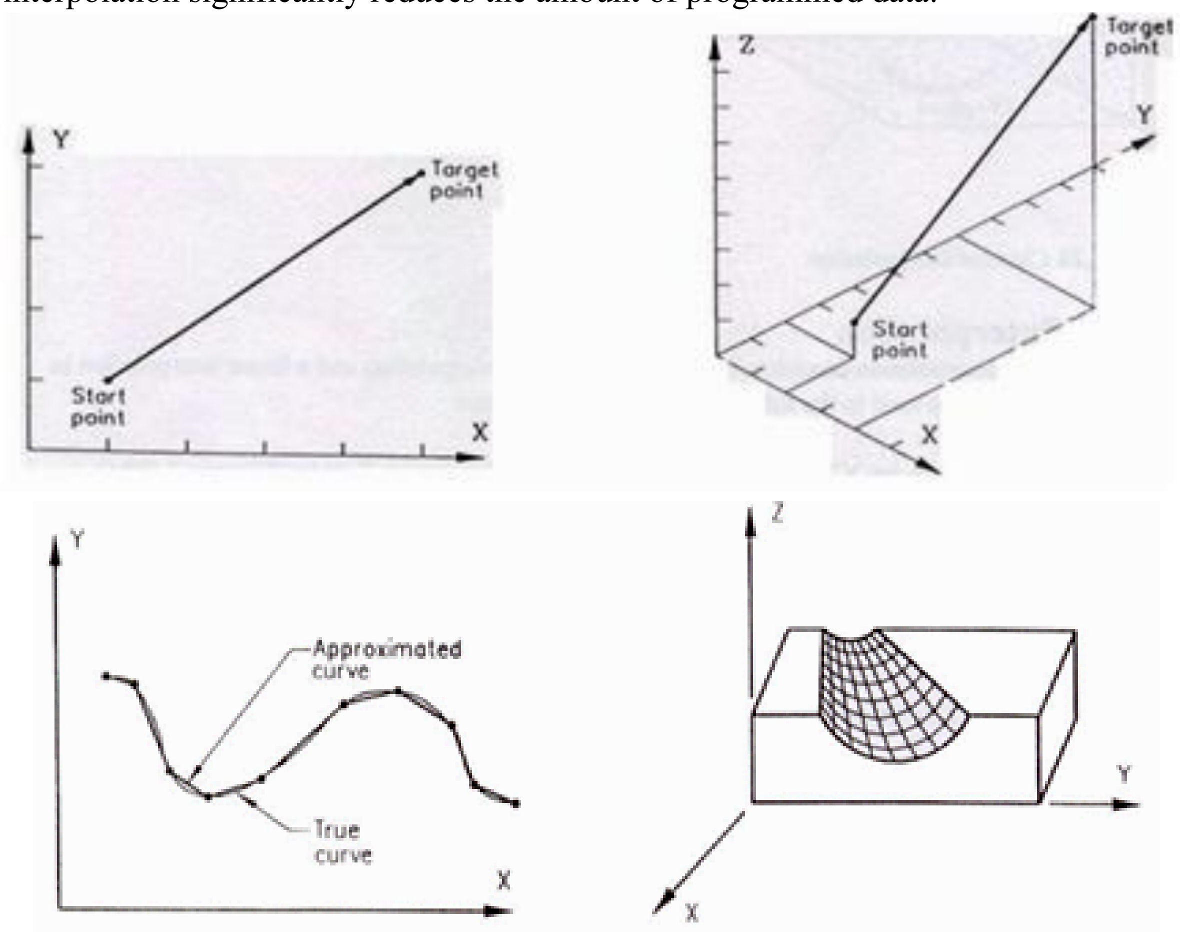


Figure 1.6 Linear Interpolation

- Implementation method:
- Example: Interpolating form point A to point E

+ Using the Digital Differential Analysis(DDA) method, consider a tool moving from point PA (starting point) to point PE(ending point) along a straight line with a defined tool speed u, as shown in the figure below.

+ Let L be the segment of the path from PA to PE. Therefore, within the time period T= L/u, the component segment $(x_E - x_A)$ and $(y_E - y_A)$ must be executed.

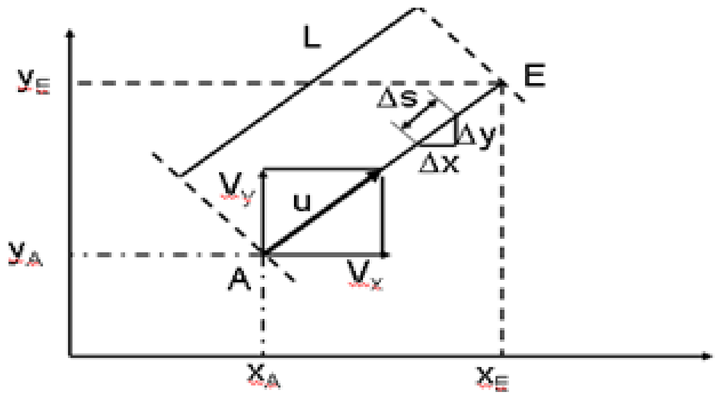


Figure 1.7 Straight Line Interpolation Method

The coordinates of intermediate point are calculated based on the time function:

$$x(t) = x_A + \int V_x dt = x_A + \int \frac{x_E - x_A}{T} dt$$

$$y(t) = y_A + \int V_y dt = y_A + \int \frac{y_E - y_A}{T} dt$$

Dividing the time T into small intervals $\Delta t = \frac{T}{N}$, the integration operation can be replaced by a summation:

$$x(t) = x(n.\Delta t) = y_A + \frac{y_B - y_A}{N}.n \quad (n = 1 \dots . N)$$

With each summation step, the position value increases by a constant step. To ensure the accuracy of the interpolation, the summation steps must be smaller than the resolution Δf of the tool drive. Typically Δf = 0, 001 mm.

$$\Delta f_{xy} \geq \max\left|\frac{x_B - x_A}{N}\right| = \max\left|\frac{y_B - y_A}{N}\right|$$

b) Circalar interpolation method:

- Concept: The Cutting tool moves from the starting point to the endpoint along a circular path with a simple command(block), replacing many straight line interpolation commands.
- Implementation: circular interpolation along 2 axes involves the following required parameters
  - Starting point coordinates, ending point coordinates, center, or radius of the circular ars, or offset I, J from the center relative to the starting point.
  - Speed of movement on each axis.
- Capability: circular interpolation of an arc a complete circle.
- Implementation method:
  - The DDA interpolation method is also applied in circular interpolation.
  - We have: PA: starting point, PE: endpoint, P: point on the curve, T: Time from PA to PE, t: Time from PA to P.
  - To run cutting along the curve, the intermediate points on the contour must be determined from the interpolation unit in a relationship dependent on the cutting time.

We have:

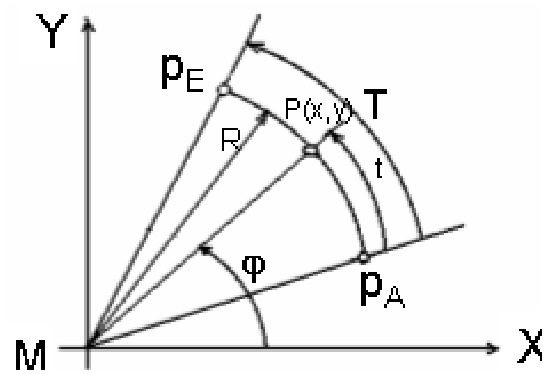


Figure 1.8 Circular Interpolation

$$x = Rcos\varphi, y = Rsin\varphi, \qquad \varphi = \frac{2\pi t}{T}$$

T: the total time to complete one full revolution

Substitute in, and we get: $x = Rcos(\frac{2\pi t}{T}), \qquad y = Rsin(\frac{2\pi t}{T})$

Taking the integral with respect to time, we obtain the component velocities in each axis separately:

$$\frac{dx}{dt} = -\frac{2R\pi}{T}\sin\left(\frac{2\pi t}{T}\right) = -\frac{2R\pi}{T}.y(t)$$
$$\frac{dy}{dt} = -\frac{2R\pi}{T}\cos\left(\frac{2\pi t}{T}\right) = -\frac{2R\pi}{T}.x(t)$$

- With sufficient accuracy, the integral operation can be replaced by the summation of displacements. Therefore:

$$x(t) = X_{PA} - \frac{2\pi}{N}\sum_{i=0}^{n} y(i\Delta t)$$
$$y(t) = X_{PA} - \frac{2\pi}{N}\sum_{i=0}^{n} x(i\Delta t)$$

1.4 Basic G-code commands:

- The Gcode command signals the star of a function. when using this command, it signifies a change in operation. There are two types of Gcode: model and non-madal.
- Modal Gcodes: They are stored in the program memory until another Gcode of the same type is called. The subsequent code will cancel the previous one in the control program.

Table 1.1 The list of G-code commands

Command Code **G**	Function
G00	Control Point Characteristics, Fast Operation
G01	Linear Interpolation.
G02	Circular Interpolation.

G03	Circular Interpolation.
G04	Dwell Time
G17	Select Coordinate Plane
G18	Select Coordinate Plane
G19	Select Coordinate Plane
G20	Unit inch
G21	Unit met
G41	Select Direction of Movement
G42	Select Direction of Movement
G56	Shift the Origin Point
G57	Shift the Origin Point
G59	Shift the Origin Point
G60	Precision Stop
G61	Precision Stop
G90	Absolute Control Data
G91	Điều khiển tương đối

CHAPTER 2 INTRODUCTION TO LASER AND APPLICATIONS

2.1 Introduction:

- Laser, an acronym for Light Amplification by Stimulated Emission of Radiation, refers to the amplification of light through forced radiation. Laser is an artificial light source obtained by amplifying light through radiation emitted when the elements of a corresponding material environment are highly activated. Laser light possesses distinct properties compared to natural or other artificial light, and it has various practical applications in many scientific, engineering, and everyday life fields, constituting a scientific and technological revolution since its inception.

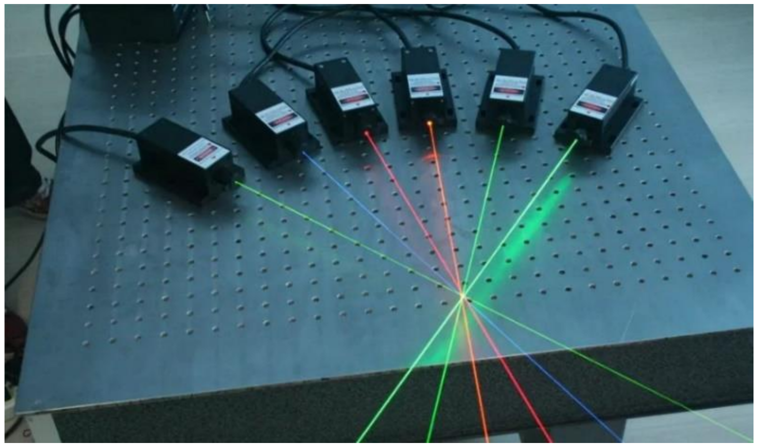


Figure 2.1 Types of Laser Heads

- The most widespread application of laser technology lies in the field of manufacturing and machining, where its capability for fast and precise cutting has propelled the industry to new heights. Unlike traditional tools, which require skilled craftsmen to invest significant time and effort in metalworking and material fabrication, laser technology addresses major drawbacks. It excels in crafting intricate details, working with extremely hard or thin materials, and handling curved or folded shapes, providing a comprehensive solution to previously insurmountable challenges.
- The application of lasers in the field of medicine has ushered in a promising new era in diagnosis and treatment. Many scientists believe that, in the 21st century, laser surgical tools will completely replace traditional surgical blades. The use of low-power lasers in non-pharmaceutical treatments also holds great promise in the field of biotechnology

2.2 Classification:

❖ Solid-state laser

- There are about 200 solid-state materials that can be used as active laser media. Some common types of solid-state lasers include
  + YAG-Neodymium: The active medium is Yttrium Aluminium Garnet (YAG) with an additional 2-5% Neodymium, emitting at a wavelength of 1060nm in the near-infrared spectrum. It can operate continuously up to 100W or pulse with a frequency of 1000-10000Hz.
  + Ruby: The active medium is aluminum crystal doped with chromium ions, emitting at a wavelength of 694.3nm in the red region of the white light spectrum.
  + Semiconductor: The most common type is Gallium Arsenide diode, emitting at a wavelength of 890nm in the near-infrared spectrum..

❖ Gas laser

  + He-Ne: The active medium is a mixture of Helium and Neon gases, emitting at a wavelength of 632.8nm in the visible light spectrum, specifically in the red region. It has low power, ranging from one to several tens of milliwatts. In medical applications, it is used as an internal laser to stimulate blood vessels.
  + Argon: The active medium is argon gas, emitting at wavelengths of 488 and 514.5nm.

+ CO_2: Wavelength of 10,600nm in the far-infrared spectrum, with emitted power reaching up to megawatts (MW). It is used in medical applications, particularly as a surgical scalpe

❖ Liquid laser

+ The active medium is a liquid, with the most common type being color lasers.

2.3 Principle of Laser Generation

❖ Structural Model.

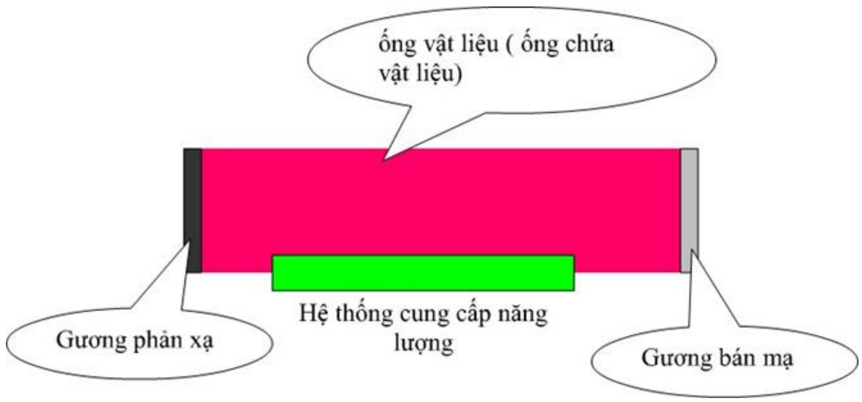


Figure 2.2 Internal Structure of the Laser Head.

- The general construction principle of a laser machine includes: the resonator chamber containing the laser active medium, the pumping source, and the optical guiding system. Among these, the resonator chamber with the laser active medium is the main component, which is a special substance capable of amplifying light through stimulated emission to generate the laser. When one photon collides with this active medium, it triggers the emission of another photon in the same direction as the incoming photon.
- On the other hand, the resonator chamber has two mirrors at both ends, one mirror completely reflects the photons when they approach, and the other allows some photons to pass through while reflecting some back, causing the photon particles to continuously collide with the laser active medium, creating a high photon density. Therefore, the intensity of the laser beam is amplified multiple times.

2.4 Mechanical of Operation:

- Lasers can be designed to operate in either continuous wave (CW) or pulsed operation modes. This leads to fundamental differences when constructing laser systems for various applications
- **Continuous wave mode**

- In continuous wave mode, the power of a laser remains relatively constant over time. The inversion of electron density necessary for laser operation is maintained continuously by a steady energy pumping source.

- **Pulsed mode**

In pulsed mode, the laser power varies over time, characterized by alternating "on" and "off" stages to concentrate the highest possible energy in the shortest time. Laser ablation is an example, where with enough energy to provide the necessary heat, lasers can vaporize a small amount of material on the surface of the sample in a very short time. However, if the same amount of energy is applied over a longer period of time, the heat will have time to penetrate deeper into the sample, resulting in less material being vaporized. There are many methods to achieve this, such as:

- Q-switching method
- Mode locking method
- Pulsed pumping method

2.5 Occupational safety and health:

- Lasers with low intensity, even just a few milliwatts, can still pose a danger to the human eye.

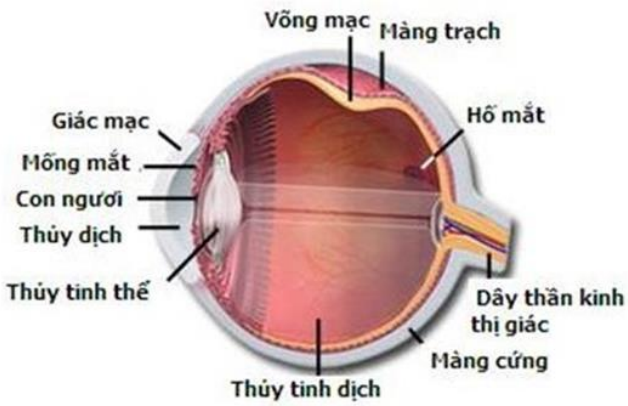


Figure 2.3 Effects of laser Beams

- At wavelengths where the cornea and crystalline lens can focus well, thanks to the coherence and high directionality of the laser, a large amount of energy can be concentrated on a very small spot on the retina.
- As a result, a focused burn destroys eye cells permanently within a few seconds, and it can even happen faster. The safety classes of lasers are categorized from I to IV. With class I, the laser beam is relatively safe. With class IV, even the diverging beam can cause damage to the eyes or skin burns. Consumer laser products such as CD players and laser pointers used in classrooms are classified as safe under categories I, II, or III.

2.6 Laser applications

Image 2.4 Laser engraving for precision applications

- At the time of its invention in 1960, the laser was hailed as the 'solution in search of applications.' Since then, lasers have become widespread, finding thousands of uses across various applications in every sector of modern society, such as in medicine, industry, scientific research, especially in the military. The laser is considered one of the most impactful inventions of the 20th century.

CHAPTER 3 MODEL DESIGN AND CALCULATION

3.1 Machine structure design:

3.1.1 Workspace

Design requiremnts:

+ Engraving area dimensions: X-axis: 320 mm, Y-axis: 350 mm, Z-axis: 150 mm, where the Z-axis is used for adjusting the focal length of the laser engraving head..
+ Maximum traverse speed on the X and Y axes: v_1 = 26 mm/s
+ The system acceleration is a = 0.5 m/s^2
+ Service lifespan: 3 years.
+ One year consists of 310 working days, and each working day is 8 hours
+ Bi-directional working load

3.1.2 Machine Dimension

- The machine dimensions: 600 x 500 x 300 mm;
- Cutting size: 320 x 350 mm.

3.1.3. Load capacity and motor selection:

- The laser cutting head directs the laser beam to cut materials without direct contact with the material. Therefore, the basic load on the transmission system is essentially zero
- For simplicity and convenience in control, and to meet technical and economic criteria, we choose stepper motors.
- Stepper motors operate on the principle that each electrical pulse causes the motor to rotate by a specific angle. The number of pulses is proportional to the displacement, and it is typically selected so that one step of the motor corresponds to a specific linear displacement..

3.1.4 Analysis and Selection of Design Options

a) Design Option 1: Moving Table

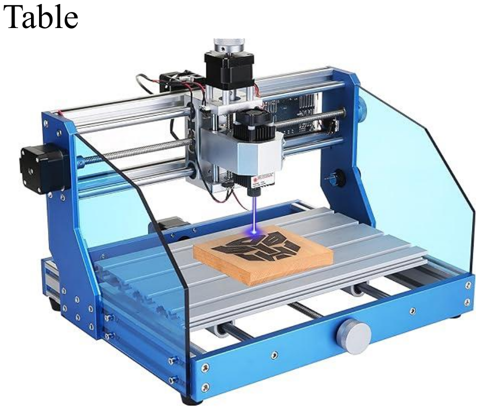


Figure 3.1 Moving machine table

- Characteristics:
  + Large mass, extensive surface area, and bulky dimensions;
  + The machine structure must be large, robust, ensuring rigidity and stability.
  + Significant vibration when the machine table is in motion.
  + Fixturing of the workpiece is required during engraving

b) Design solution 2: Stationary machine table

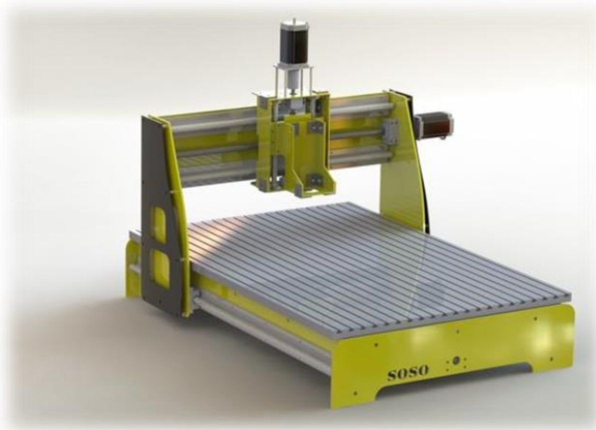


Figure 3.2 Stationary Machine Table

- This type of machine table, when cutting, experiences less load on the cutting head. These machines typically have large travel distances and broad surface coverage. In laser engraving and cutting machines, comparing the movement of the machine table to that of the laser cutting head, the stationary table option is much more efficient.
- The cutting head is a laser beam box that emits the laser beam only when in operation. During movement, it imposes minimal load on the transmission system, allowing for fast, efficient, and energy-saving motion.

Characteristic:
  + The machine structure becomes simpler and less bulky, as it does not require high rigidity and load-bearing capacity..
  + The manufacturing cost for the stationary machine table option is lower compared to the moving machine table.

Conclusion: From the above analysis, it is evident that, for a stationary machine table, having the laser cutting head in motion is the most feasible and cost-effective option when designing a laser engraving machine

3.1.5 Drive System Options Selection

a) Screw and Nut Transmission System

Advantages:

- This transmission system reduces friction between the ball screw and the nut, enhancing the precision of motion. To ensure smooth rolling friction, the balls need to move continuously through a track of 4 grooves, guiding the balls from the end groove of the screw to the head groove.
- It typically offers higher accuracy compared to lead screw and nut systems.

Disadvantages:
- High cost

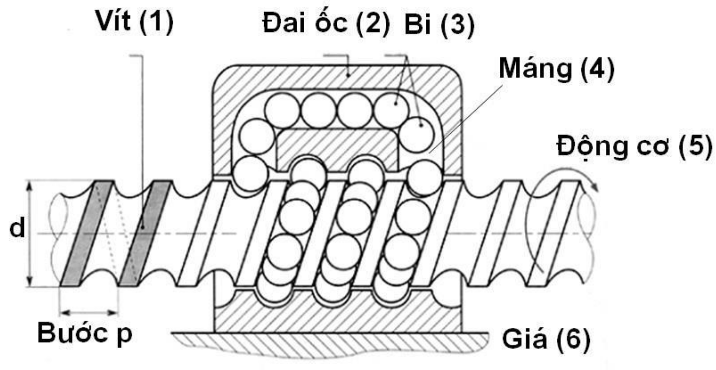


Figure 3.3 Screw and nut- ball screw

b) Belt Transmission System:

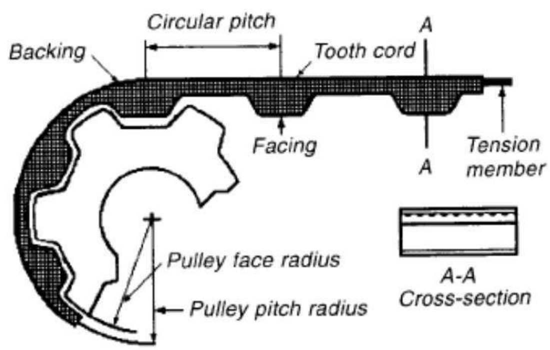


Figure 3.4 Timing belt transmission system

Advantages of the transmission system:
+ Simple structure, cost-effective.
+ "Smooth operation, noise-free due to the flexibility of the belt, suitable for high-speed transmission."
+ Transmission between distant axes (with a maximum working distance of up to 95mm is entirely feasible and practical for belt drive)
+ High transmission efficiency
+ Convenient and easy to control.

Disadvantages:
+ Size and dimension issues may arise, especially for large frames.
+ Significant force acts on the shaft bearing, and initial belt tension is required
+ Limited lifespan of the belt
+ Requires high precision, making assembly and installation more challenging
+ Difficulty in adjusting the parallelism of the belt surface for precise axis alignment

Conclusion: Based on the analysis of the pros and cons, combined with considerations of component availability and economic conditions, the belt drive system is chosen for the X-axis, and the ball screw for the Y-axis. For the Z-axis, which does not require high precision, a regular lead screw is employed for height adjustment.

3.2 Mechanical Design Considerations

+ 3D Models of Components and Details

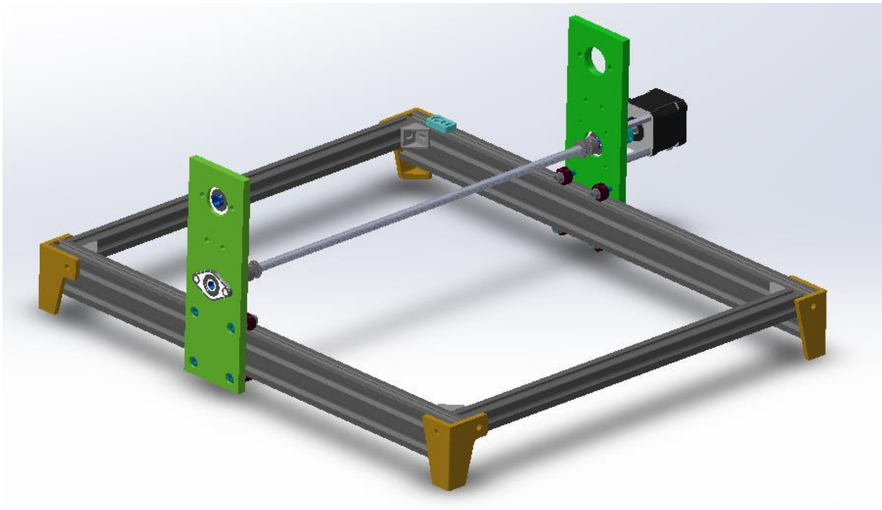


Figure 3.5 X-Axis Assembly

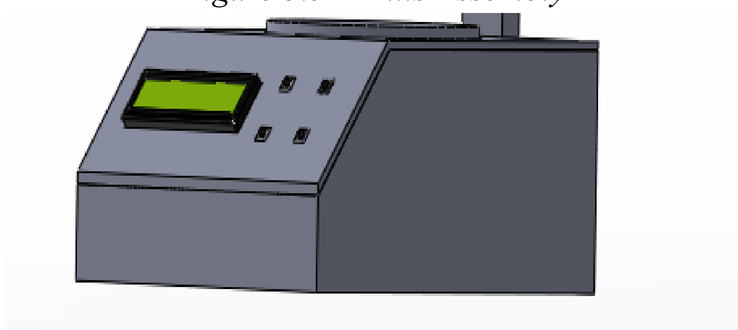


Figure 3.6 Machine Control Electrical Cabinet

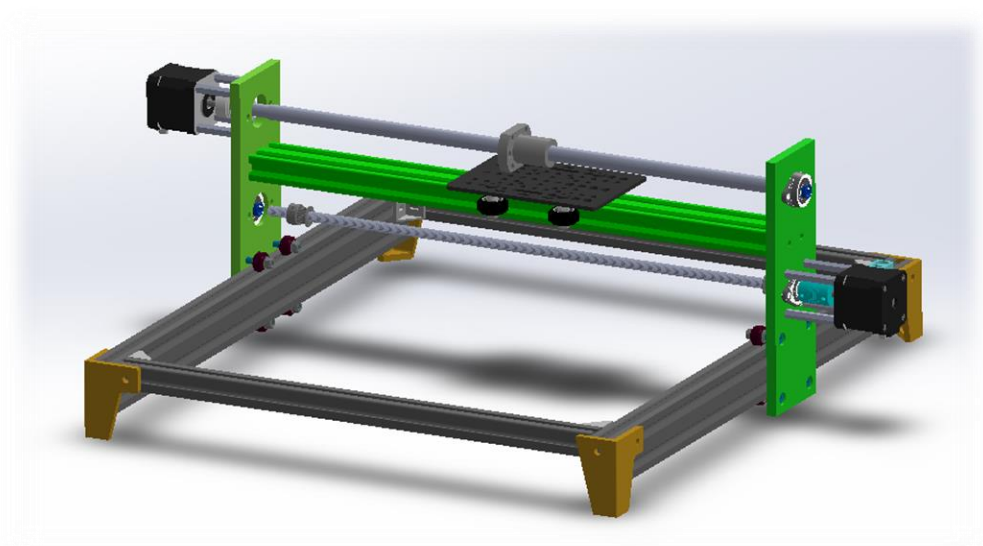


Figure 3.7 Y-Axis Assembly

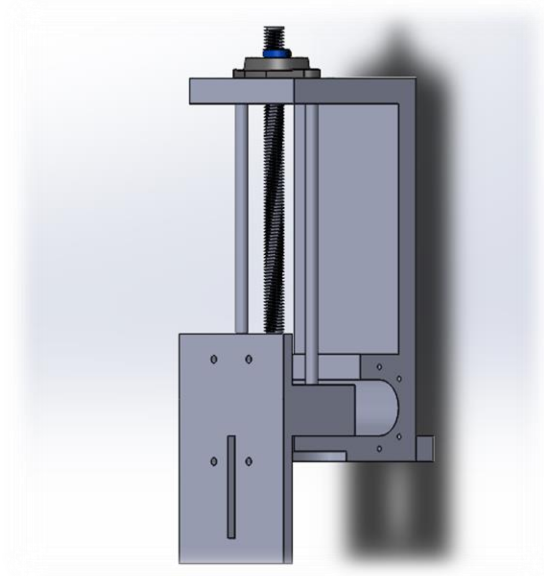


Figure 3.8 Z-Axis Lifting Assembly

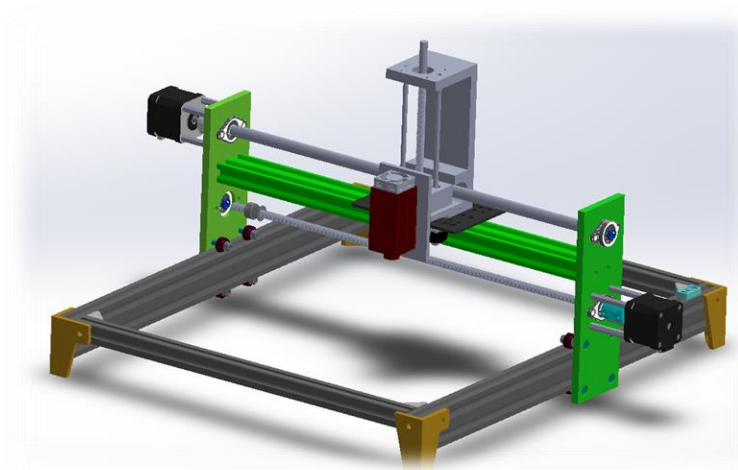


Figure 3.9 Complete Assembly Model

3.3 X-Axis Mechanism:

a) Block Diagram**:**

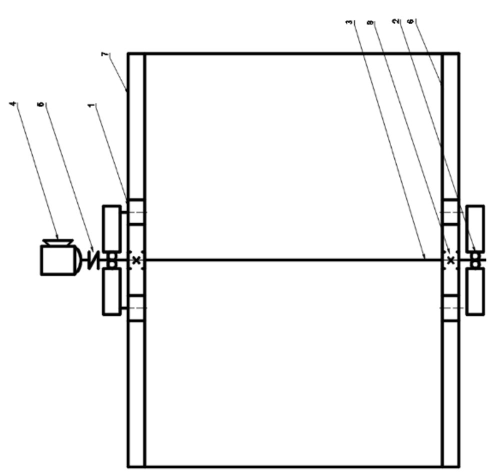


Figure 3.10 Dynamic diagram of X-Axis

Notes:

1 – Roller	4 – Motor	7 – Guide Rail
2 – Bearing	5 – Shaft Coupling	8 – Puly
3 – Drive shaft	6 – Belt	

Motion Principle Diagram

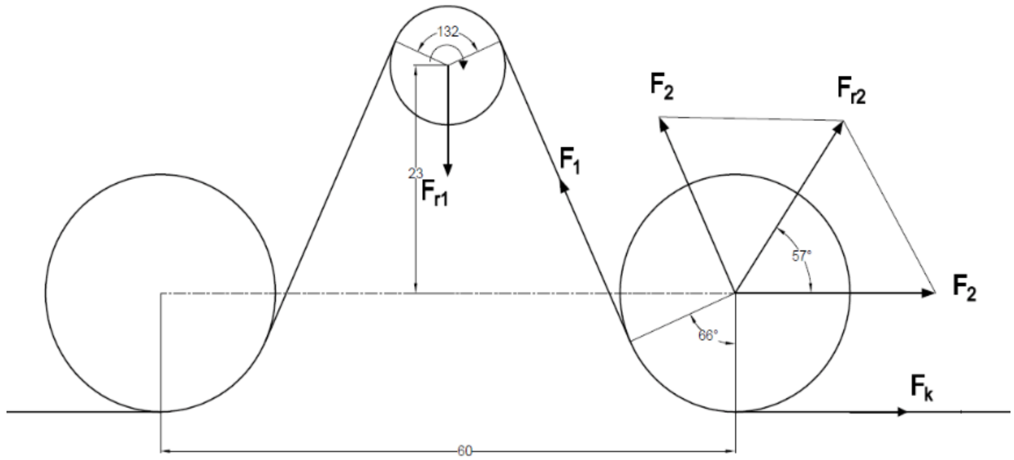


Figure 3.9 Force Analysis Diagram on Roller and Shaft

b) Calculation of the Drive Assembly on the X-Axis

- Considering the combined mass of both the Y-axis and Z-axis, exerting force on the 4 rollers (2 on the left and 2 on the right) with their respective masses m = $m_Y + m_Z = 4 + 2 = 6\ kg$, So, each roller will bear the load with $m_{conlan} = \frac{6}{4} =$ 1,5 kg thus distributing the load. F = 1,5.9,8 = 14,7 N.
- The input parameters are:
  + The gravitational force acting on each wheel: F = 14,7 N
  + Inclination angle α_1 = 132^0

Parameters puly GT2

+ Number of teeth: 20 teeth
+ Tooth pitch puly: p = 2 mm
+ Belt Width: 7 mm
+ Outer Diameter: 16 mm
+ Gear Diamete: d_{puly} = 12 mm (puly)
+ Height h = 16 mm
+ Material: Aluminum

Belt

+ Belt Pitch p = 2 mm
+ Belt Width: b = 6 mm
+ Belt Thickness: $\delta = 1{,}38\ mm$

Roller (wheel)

+ Large Pulley (Roller) Diameter d_{wheel} = 24 mm

- Angular velocity of the driven pulley (roller)

$$\omega_{\text{wheel}} = \frac{2v_{\text{wheel}}}{d_{wheel}} = \frac{2.26}{24} = 2{,}2\, rad/s$$

Where $v_{\text{wheel}} = v = 26$ mm/s speed of the X-axis movement

- Transmission Ratio

$$i = \frac{d_{wheel}}{d_{puly}} = \frac{\omega_{puly}}{\omega_{wheel}} \qquad TL[2]$$

We have the actual transmission ratio.

$$i = \frac{d_{wheel}}{d_{puly}} = \frac{24}{12} = 2$$

Angular velocity of the driving pulley (roller).

$$\omega_{puly} = \omega_{wheel}.i = 2{,}2.2 = 4{,}4\, rad/s$$

- Rotational speed on the shaft

$$n_{puly} = \frac{60\omega_{puly}}{2\pi} = \frac{60.4{,}4}{2\pi} = 42 \quad (\text{revolution/minutes}) \quad \leq n_{\text{đc}}$$

- Required motor power.

$$N_{ct} = 4.\frac{\text{F . v}}{\eta} = 4.\frac{14{,}7.0{,}026}{1.0{,}9925.0{,}97} = 1{,}6\, W < N_{\text{đc}}$$

From Table 2-1, page 27, document [1], we obtain the following data::

$\eta_{\text{đ}}$ = 0,97 - Efficiency of the belt drive system

η_{ol} = 0,9925 - Efficiency of a pair of rollers

η_k = 1 - Efficiency of the coupling

- Selecting Nema 17 STP – 42D4016-01 Stepper Motor
  + Operating voltage 12,6 V
  + Rated current 1A
  + Maximum torque $M_x = 320$ mN.m
  + Stepper Motor $1{,}8^0$
  + Motor speed 360 r/min
- Engine capacity:

$$N_{dc} = 2\pi.M_x.\frac{N}{60} = 2\pi.0{,}32.\frac{360}{60} = 12\, W > N_{ct} \text{ (Satisfy)}$$

- Force acting on the pulley shaft - Reference [2], page 72
  - Initial tension (for the timing belt drive) is applied to overcome gaps during meshing and ensure good contact between the belt and pulley. It just needs to be greater than the centrifugal force produced:

$$F_0 = (1{,}1 \div 1{,}3)F_{\text{v}} = (1{,}1 \div 1{,}3)q_m b v^2 \qquad TL[2] - \text{page } 72$$

  + In there $F_v = q_m b v^2$ is the tension caused by centrifugal force

q_m is the mass per meter length of the belt (value qm=0.0032 from Table 4.31 -TL[2])

b = 6 mm belt width

Therefore

$$F_0 = (1{,}1 \div 1{,}3)q_m b v^2 = (1{,}1 \div 1{,}3).0{,}0032.0{,}006.360^2 = 3{,}3\ N$$

Belt speed

$$\mathrm{v}_1 = \frac{\pi d_1 n}{6000} = \pi.\frac{12.360}{60000} = 0{,}23\ m/s$$

- Torque F_t

$$F_t = \frac{\mathrm{P}}{\mathrm{v}_1} = \frac{1{,}6}{0{,}23} = 6{,}96\ N$$

- Force acting on the shaft

$$F_r = (1{,}0 \div 1{,}2)F_t = (1{,}0 \div 1{,}2).6{,}96 = 8{,}4\ N \qquad \mathrm{TL[2] - page\ 72}$$

c) Calculate shaft diameter and select bearing

- Preliminary shaft diameter

$$d \geq C\sqrt[3]{\frac{N}{n}}$$

In there: d - Shaft diameter. (mm)

C – The allowable torsional stress factor for the shaft and the transmission shaft combined is taken as C = 120 as described on page 115 (TL[1])

N – Power transmission, kW

n – Shaft revolutions per minute (RPM)

Substitute the given data above to obtain.

$$d \geq C\sqrt[3]{\frac{N}{n}} = 120.\sqrt[3]{\frac{12.10^{-3}}{360}} = 3{,}86\ mm$$

⇨ Sellect d = 8 mm

- Approximate calculation
- Forces acting on the shaft
  + Centrifugal force $F_r = F_{r1} = F_{r2} = 8{,}4\ N$
  + Torque $F_t = F_{t1} = F_{t2} = 6{,}96\ N$
  + Shaft length l = 450 mm

Length of each shaft section by drawing a structural diagram

k = 15 mm , l = 420 mm, m = 15 mm

Calculate reactions at the shaft bearings.

Considering YOZ

- $\sum m_{C_y} = -F_{r1}.k - F_{r2}.(k+m) + R_{Dy}.(k+m+n) = 0$

⇨ $R_{Dy} = \frac{F_{r1}.k + F_{r2}.(k+m)}{m+n+k} = \frac{8{,}4.15 + 8{,}4.(15+420)}{450} = 8{,}4\ N$

- $\sum Y = R_{cy} - F_{r1} - F_{r2} + R_{Dy} = 0$

⇨ $R_{Cy} = F_{r1} + F_{r2} - R_{Dy} = 16{,}8 - 8{,}4 = 8{,}4\ N$

Considering XOZ

- $\sum m_{C_x} = -F_{t1}.k - F_{t2}.(k+m) - R_{Dx}.(k+m+n) = 0$

$$\Rightarrow\ R_{Dx} = \frac{-F_{t1}.15 - F_{t2}.435}{450} = \frac{-6{,}96.15 - 6{,}96.435}{450} = -6{,}96\ N$$

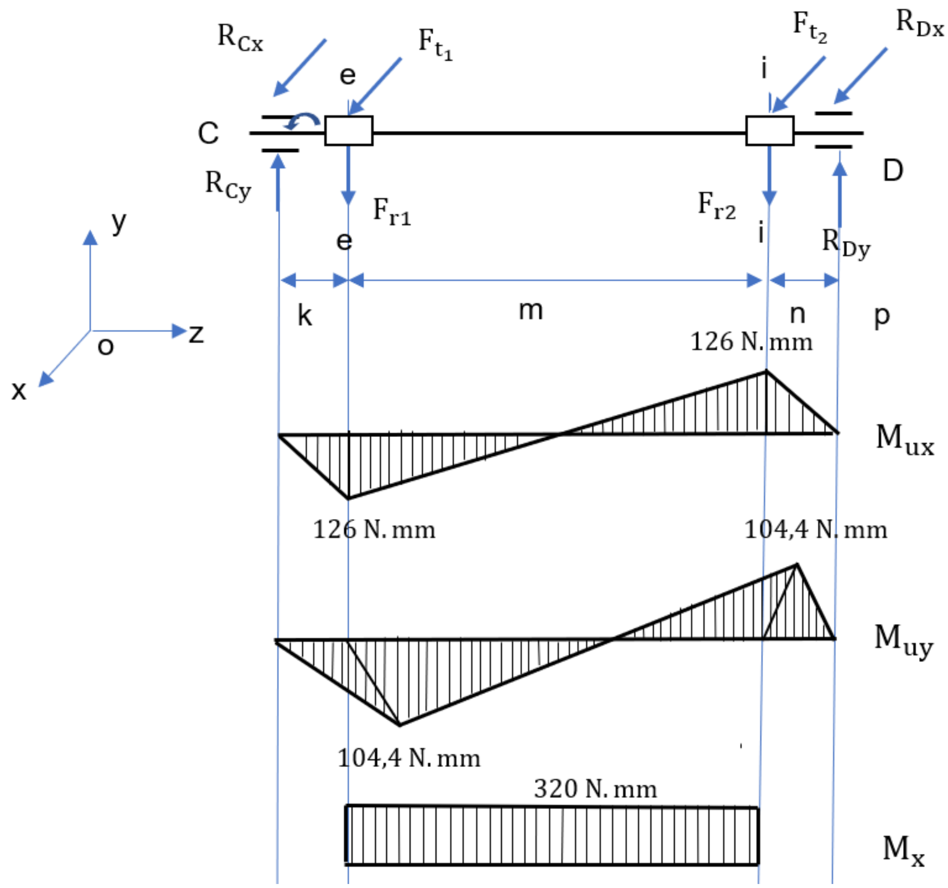


Figure 3.11 X-axis shaft moment chart

It follows that $R_{Dx}$is opposite to the chosen direction.

$$\sum X = R_{Cx} + F_{t1} + F_{t2} + R_{Dx} + P_k = 0$$

$$\Rightarrow\ R_{Cx} = -F_{t1} - F_{t2} - R_{Dx} = -6{,}96 - 6{,}96 + 6{,}96 = -6{,}96\ N$$

It follows that R_Cx is opposite to the chosen direction

- Calculate bending moments at critical sections on the X-axis
  - At section e-e

$$M_{u\,e-e} = \sqrt{M_{uy}^2 + M_{ux}^2}\quad (1)$$

In there

$$M_{ux} = R_{Cy}.k = 8{,}4.15 = 126\ \ N.mm$$
$$M_{uy} = R_{Cx}\,.k = 6{,}96.15 = 104{,}4\ N.mm$$
$$M_{u\,e-e} = \sqrt{126^2 + 104{,},4^2}\ = 164\ N.mm$$

  - At section i-i

$$M_{u\,i-i} = \sqrt{M_{uy}^2 + M_{ux}^2}\quad (*)$$

In there

$M_{ux} = -R_{Dy}.n = -8{,}4.15 = -126\ N.mm$

$M_{uy} = -R_{Dx}.n = -6{,}96.15 = -$ 104,4 N.mm

Substitute into (*) to get: $M_{u\,i-i} = \sqrt{(-126)^2 + (-104{,}4)^2}\ = 164\ N.mm$

- Calculate the shaft diameter at the critical section using formula (7-3) TL[1]

$$d \geq \sqrt[3]{\frac{M_{td}}{0{,}1(1-\beta^4)[\sigma]}} \qquad \text{(Formula } (7-4) page\ 117 \text{ TL[1])}$$

With $M_{td} = \sqrt{M_u^2 + 0{,}75M_x^2} = \sqrt{164^2 + 0{,}75.320^2}\ =\ 322\ N.mm$

In there: M_{td} is the equivalent moment

M_x, M_u are the torsional and bending moments at the calculated section, N.mm

Ratio $\beta = \frac{d_0}{d}$ = 0 (As the shaft is not hollow)

Allowable stress $[\sigma]$ = 50 $N/mm2$

Substitute into the formula:

$$d \geq \sqrt[3]{\frac{M_{td}}{0{,}1(1-\beta^4)[\sigma]}} = \sqrt[3]{\frac{322}{0{,}1(1-0^4).50}} = 4\ mm$$

⇨ d = 8 mm

d) Choosing roller bearings

We select a type of ball bearing for both the X and Y axes due to the light working conditions, minimal vibration, and low axial loads.

Initial data::

+ Rotation speed of the X-axis n = 42 revolutions/minute
+ Shaft diameter d = 8 mm
+ Working time h = 3.310.21= 7440 hour

We have:

$$R_C = \sqrt{R_{Cy}^2 + R_{Cx}^2} = \sqrt{8{,}4^2 + -6{,}96^2} = 11\ N$$

$$R_D = \sqrt{R_{Dy}^2 + R_{Dx}^2} = \sqrt{-6{,}96^2 + 8{,}4^2} = 11\ N$$

Selection diagram for the X- axis bearing:

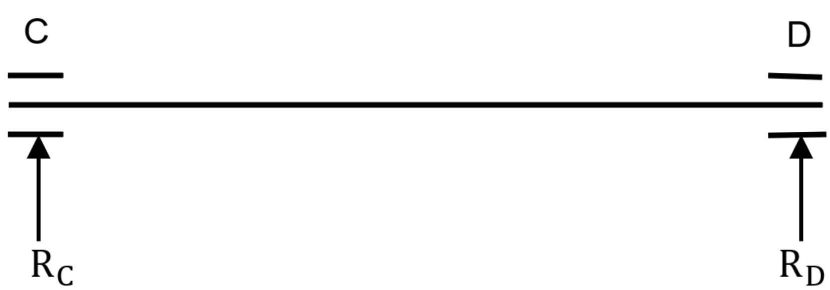


Figure 3.12 Diagram of forces acting on the roller bearing

Efficiency factor calculated using formula (8-1) TL[1]:

$$C = Q(nh)^{0,3} \leq C_{table} \quad TL[1]$$

In there

$Q = (K_v R + mA)K_n K_t$ – Equivalent load formula (8-7) TL[1]

Since there is no axial load, $A = 0$

Calculate Q for the E bearing as it has a large load R_E

$K_t = 1$ – Static and Impact Load Factor, from table (8-3) TL[1]

$K_n = 1$ – Operating temperature below 100°C, from table (8-4) TL[1]

$K_v = 1$ – Inner ring rotation factor, from table (8-5) TL[1]

Since the radial load at the C bearing is equal to that at the D bearing, we only calculate for the C bearing and choose a bearing for this shaft. The D bearing will be of the same type..

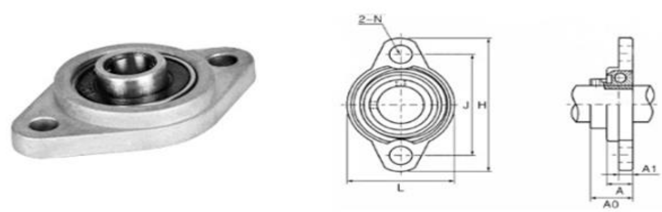
Substitute values:

$$Q = K_v . R_E . K_n . K_t = 11.1 = 11\ daN$$

Calculate C according to the formula(8-1) on $C = 11.(42.7440)^{0,3} = 489,6 \leq C_{table}$

From table 14 P TL [1] orresponding to d = 8 mm Select bearing type 18 (special type, light and medium duty).

Based on available information, we select the following type of roller bearing:



带座轴承代号 Bearing Unit No	轴承 Shaft Dia d (mm)	外形尺寸(毫米)Dimensions (mm) H	J	A1	A	N	L	A0	螺栓 Bolt Used	重量 Weight
KFL08	8	48	36	4	8.5	5	27	12	M5	26g
KFL000	10	60	45	5.5	11.5	7	36	15.5	M6	38g
KFL001	12	63	48	5.5	11.5	7	38	16	M6	42g

Figure 3.13 Catalog specifications for roller bearing KFL08.

Table 1.2 Roller Bearing Specifications

Symbol	d (mm)	D (mm)	B (mm)	Ball Diameter	C_{table}
18	8	22	7	3,97	3700

e) Calculating Forces on the Roller

- Radial force of the roller

$$F_{rl} = 2F_0 Sin\left(\frac{\alpha_2}{2}\right) = 2.3,3.\sin\left(\frac{66}{2}\right) = 3,6\ \ N \quad TL[4]$$

- Friction force on the roller:

$$F_{ms} = P.\mu = 14,7.0,1 = 1,47\ N$$

With μ Coefficient of friction between the roller and the belt.

- Moving force on the roller

$F_{cd} = F_{rl}.Cos(57^0) = 3,6.\cos(57^0) = 2\ N\ >\ F_{ms}$ (meets the requirements)

f) Calculate the steps of the movement: here, the belt is designed to move linearly by 1 mm.

$$b = \frac{360}{\alpha.\lambda.R.m} = \frac{360}{1{,}8^0.2.20.\frac{1}{16}} = 80\ step/mm$$

With $\alpha = 1{,}8^0$ is the step angle, $m = \frac{1}{16}$ is the microstepping mode when using A4988, R is the number of teeth on the puly, $\lambda = 2$ is the pitch of the puly.

3.4 Calculate parameters for the Y-axis.

Design parameters:

+ Maximum speed v = 26 mm/s
+ Maximum operating acceleration of the system a = 0,5 m/s^2
+ Mass of the Y axis load is 2 kg

a) Design the lead screw:

- Calculate the axial force

$$F_a = F_M + F_f + F_i + F_g \qquad TL[6]$$

With:

+ F_M = 0 machining force
+ $F_f = mg\mu cos\theta = 2.9{,}8.0{,}05.\cos 0 = 0{,}98$ N
+ $F_g = m.g.sin\theta = 0$ N
+ $F_i = m.a = 2.0{,}5 = 1$ N

→ $F_a = 1{,}98$ N

- Choose the mounting type: Fixed-free type, where one end of the lead screw is fixed, and the other end is free..

Calculate the lead screw motion parameters according TL[2]

+ Choose material: C45 steel for the lead screw, steel for the nut, and use right-hand threads

We have a force along the axis F_a acting on the screw shaft transmission system.

$$F_a = 1{,}98\ N$$

Calculate the average diameter of the thread:

$$d_1 \geq \sqrt{\frac{4.1{,}3F_a}{\pi[\sigma_k]}} \qquad TL[4]$$

In there: F_a is a force along the axis

$[\sigma_k] = [\sigma_n] = \frac{\sigma_{ch}}{3} = \frac{360}{3} = 120$ σ_{ch} being the yield strength of the material of the ball screw

Substitute the values:

$$d_1 \geq \sqrt{\frac{4.1{,}3.1{,}98}{\pi.120}} = 0{,}165\ mm$$

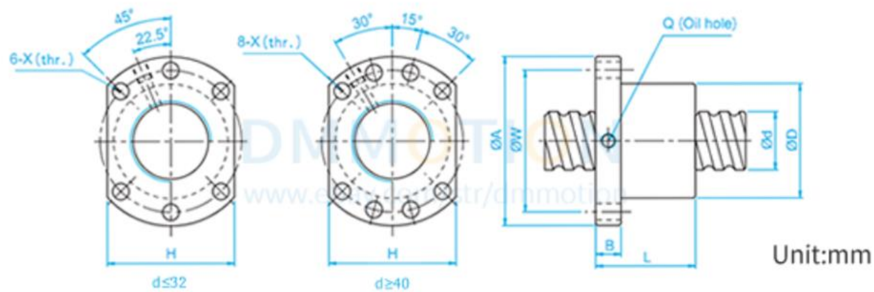
+ Based on the data above combined with the market reality, we choose:

=> Selecting the ball screw SFU 1204 -4 with $d_1 =$ 12 mm, lead l = 4 mm We look up in the table above

Ball Screw Parameters:

Table 1.3 Ball Screw 1204 Parameters

Symbol	D	A	B	W	C_a	C_0
SFU-1204	22	42	8	32	400	670



Moder No.	d	l	Ball Dia	Dimension(mm)									Rating Load(Kgf)	
				D	A	B	L	W	H	X	Q	n	Dynamic	Static
SFU1204-3	12	4	2.381	22	42	8	35	32	4.5	30	M6	3	400	670
SFU1605-3	16	4	3.175	28	48	10	42	38	5.5	40	M6	3	765	1320

Figure 3.14 Catalog vít me bi SFU1204-4

b) Calculate the motor selection for the Y-axis:

Select a preliminary motor:

- Y-axis input parameters:

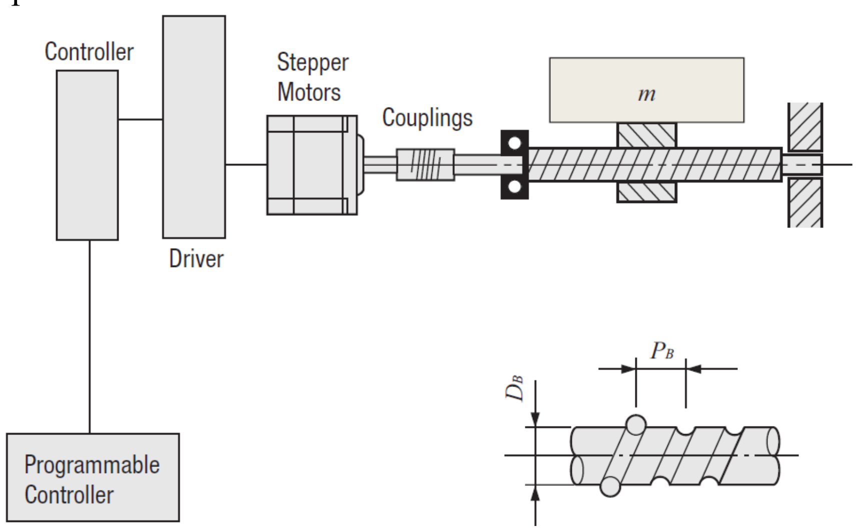


Figure 3.15 Load Distribution Diagram on the Ball Screw

Table 1.4: Y-Axis Parameters

Data	Parameters
Stroke length l_y	$l_y = 350\ mm$
Load Mass	$m_y = 2kg$
Coefficient of Friction between Wheel and Rail	$\mu = 0.05$
Efficiency of the Ball Screw	$\eta = 0{,}95$
Coefficient of Friction between the Screw and Nut.	$\mu_0 = 0.1$

Diameter of the Ball Screw.	$D_B = 12\ mm$
Lead of the Ball Screw (per revolution)	$P_B = 4$ mm
Time for one motor pulse to drive one step	$t_0 = 0{,}8\ s$
Material of the Ball Screw:	Steel ($\rho = 7{,}9 \text{ x } 10^3\ kg/m^3$)
Safety Factor	$S_f = 2$
Required Resolution (screw pitch/motor resolution)	$\Delta l = 0.02\ mm/step$

+ Required Resolution of the Stepper Motor

$$\theta_s = \frac{360.\Delta L}{P_B} = \frac{360.0.02}{4} = 1{,}8^0$$

+ Total number of steps A_x in the operating range

$$A_x = \frac{l_x.360}{D_B.\theta_s} = \frac{350.360}{12.1{,}8} = 5833\ (bước)$$

+ Acceleration time (t1):

$$t_1 = t_0.0{,}25 = 0{,}8.0{,}25 = 0{,}2\ s$$

+ Operating frequency f_2:

$$f_2 = \frac{A_x - f_1.t_1}{t_0 - t_1} = \frac{5833 - 0.0{,}6}{0{,}8 - 0{,}2} = 9{,}72\ (kHz)$$

+ Operating Speed

$$N_m = f_2 \frac{\theta_s}{360}.60 = 9{,}72.\frac{1{,}8}{360}.60.10^3 = 2916\ (r/min)$$

Force induced by moving load.

$$F = F_A + m_y.g.(sin\theta + \mu.cos\theta) = 0 + 2.9{,}8(sin0 + cos0.0{,}05)$$
$$= 0{,}98\ N$$

+ Calculate the Load Moment $T_L\ (N.m)$

Force exerted on the ball screw: $F_0 = \frac{1}{3}F = 0{,}327\ N$

Therefore:

$$T_L = \left(\frac{F.P_B}{2\pi.\eta} + \frac{F_0.\mu_0.P_B}{2\pi}\right)\frac{1}{i} = \frac{0{,}98.4 \text{ x } 10^{-3}}{2\pi.0{,}95} + \frac{0{,}327\ .4 \text{ x } 10^{-3}\ .0{,}1}{2\pi}.1$$
$$= 6{,}6775 \text{ x } 10^{-4}\ [N.m]$$

+ Calculate the Moment of Inertia

Inertia Moment of the Ball Screw:

$$\mathrm{J_B} = \frac{\pi}{32}.\rho.L_B.D_B^4 = \frac{\pi}{32}.7{,}9 \text{ x } 10^3.490.10^{-3}.(12 \text{ x } 10^{-3})^4$$
$$= 7{,}88 \text{ x } 10^{-6}\ kg.m^2$$

Inertia Moment of the Load:

$$\mathrm{J_r} = m\left(\frac{P_B}{2\pi}\right)^2 = 2 * \left(4 * \frac{10^{-3}}{2\pi}\right)^2 = 8{,}11 * 10^{-7}\ kg.m^2$$

+ Total Moment of Inertia

$$J_L = J_B + J_r = (7{,}88 + 8{,}11) * 10^{-7} = 1{,}599 \text{ x } 10^{-6} \; kg.m^2$$

+ Angular Acceleration T_a $[N.m]$

$$T_a = \frac{(J_0 + J_{Ly})}{9{,}55} . \frac{N_M}{t_1} = \frac{J_0 + 1{,}599 \text{ x } 10^{-6}}{9{,}55} . \frac{2916}{0{,}2} = 1526\, J_0 + 2{,}44 \text{ x } 10^{-3} \; [N.m]$$

+ Calculate the required torque T_M $[N.m]$:

With a safety factor $S_f = 2$ therefore:

$$T_M = S_f(T_L + T_a) = 2(\,1526\, J_0 + 2{,}44 \text{ x } 10^{-3} + 6{,}6775 \text{ x } 10^{-4}\,)$$

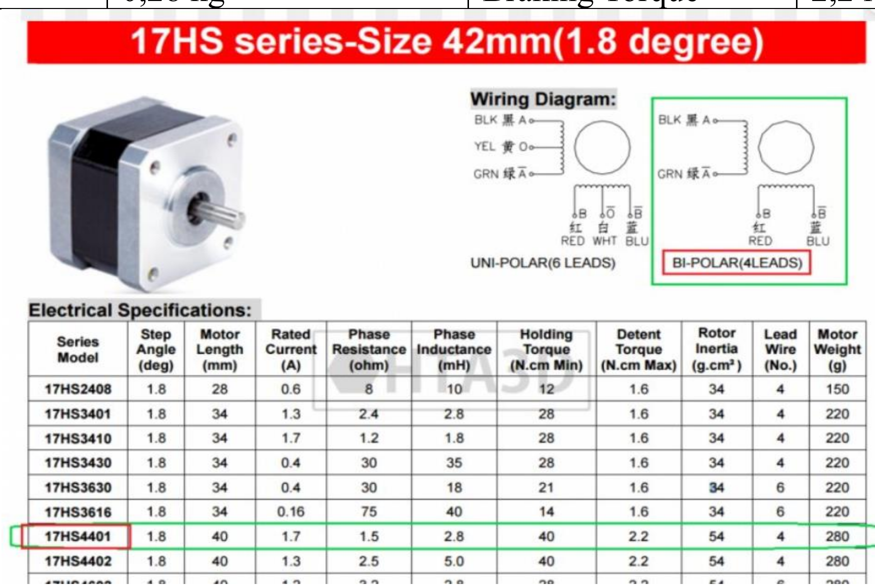
$$= 3052\, J_0 + 9{,}12 \text{ x } 10^{-3} \; [N.m]$$

From the calculated values above, we can choose the following motor:

Technical Specifications :

Table 1.5: To select a motor with technical specifications

Stepper motor Nema 17			
General Technical Characteristics		Electrical Technical Characteristics	
Step Angle	$1{,}8^0$	Voltage Rating	4,42 V
Number of Phases	2	Current Intensity	1,7 A
Thermal Resistance	$100 M\Omega$ min $(500 VDC)$	Phase Resistance	1,5 Ω
Insulation Type	B	Phase Impedance	2,8 mH
Rotor Inertia	$54\; g \text{ x } cm^2$	Torque	40 N.cm
Mass.	0,28 kg	Braking Torque	2,2 N.cm



Series Model	Step Angle (deg)	Motor Length (mm)	Rated Current (A)	Phase Resistance (ohm)	Phase Inductance (mH)	Holding Torque (N.cm Min)	Detent Torque (N.cm Max)	Rotor Inertia (g.cm²)	Lead Wire (No.)	Motor Weight (g)
17HS2408	1.8	28	0.6	8	10	12	1.6	34	4	150
17HS3401	1.8	34	1.3	2.4	2.8	28	1.6	34	4	220
17HS3410	1.8	34	1.7	1.2	1.8	28	1.6	34	4	220
17HS3430	1.8	34	0.4	30	35	28	1.6	34	4	220
17HS3630	1.8	34	0.4	30	18	21	1.6	34	6	220
17HS3616	1.8	34	0.16	75	40	14	1.6	34	6	220
17HS4401	1.8	40	1.7	1.5	2.8	40	2.2	54	4	280
17HS4402	1.8	40	1.3	2.5	5.0	40	2.2	54	4	280

Figure 3.16 Nema17 – 17HS4401 Stepper Motor Specifications

With $J_0 = 54 \text{ x } 10^{-7} \; kg.m^2$ In that case, the required torque according to the motor specifications is:

- Testing the required torque moment:

$$T_{Mz} = 3052\, J_0 + 9{,}12 \text{ x } 10^{-3} = 3052\,.54 \text{ x } 10^{-7} + 9{,}12 \text{ x } 10^{-3} \approx 0{,}03 < 0{,}4\; N.m$$

3.5 Shaft Z Design Calculation Parameters

- Used to adjust the focus and distance of the laser head for different materials.

a) Parameters**:**

- Input parameters:
  + Mass of laser head: 200 g
  + Working length: 100 mm
  + Transmission function: used to move the laser up and down to adjust the focus
  + Working conditions: normal environment

b) Calculation and selection of the screw - nut

- Axial force

$$F_a = F_M + F_f + F_i + F_g \qquad TL[6]$$

With:

+ $F_M = 0$ is the machining force.
+ $F_f = mg\mu cos\theta = 0{,}2.9{,}8.0{,}05.\cos 90 = 0$ N
+ $F_g = m.g.sin\theta = 0$ N
+ $F_i = m.a = 0{,}2.0{,}5 = 0{,}1$ N

→ Therefore: $F_a = 0{,}1$ N

- Calculate the motion screw parameters - refer to Reference [2]

- Choose the material for the screw - steel 45, and for the nut - brass rod, using a square thread with a right-hand thread.
  + The average diameter of the thread:

$$d_2 \geq \sqrt{\frac{F_a}{\pi\psi_h\psi_H[q]}} \qquad TL[2]$$

In there: F_a is the axial force

$\psi_h = 0{,}5$ is the height factor for square threads

$\psi_H = 1{,}2 \ldots 2{,}5$ s the height factor for the entire nut

[q] is the permissible pressure

- The material for the screw and nut is steel - brass, so we choose $\psi_H = 2$ and [q] = 9 (Mpa).

$$d_2 \geq \sqrt{\frac{0{,}1}{\pi.1{,}2.0{,}5.9}} = 6 \text{ x } 10^{-3}\; mm$$

=> Choose $d_2 = 7$ mm, Inside diameter. $d_1 = 6$ mm, Thread pitch p = 2 mm and outer diameter d = 8 mm, refer to table P2.4 page 251 - TL[2]

- Based on the above data combined with market reality, we choose a type of screw named: Tr8x2

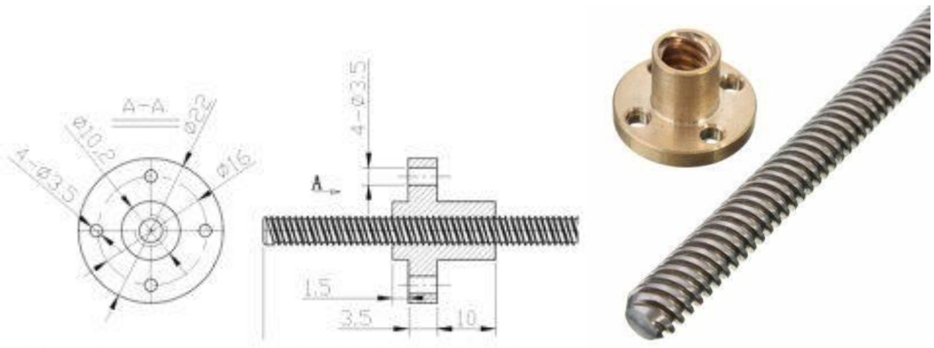


Figure 3.17 Lead Screw T8

Model	d(mm)	Pitch	L	D	B	A	W
Tr8x2	8	2	15	10,2	3,5	22	16

Table 1.6: Screw parameters table

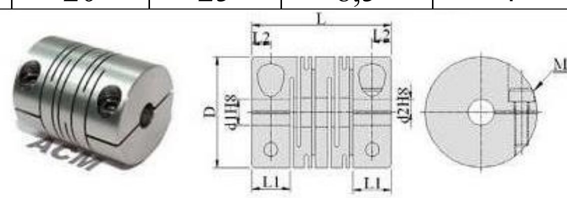
3.6 Coupling

- We use a flexible coupling. The two halves of the coupling are connected by a flexible part, which can be made of metal or rubber.
  + The material for the shaft is forged steel 35, and the material for the pin is normalized steel 45.
  + The flexible coupling is capable of absorbing impacts and vibrations. It has a simple structure, easy to manufacture, and is cost-effective, making it widely used.

- Torsional moment of the motor: M_x= 430 N.mm = 0,43 $N.m$ < 1,5 N.m (satisfactory)

Table 3.6 Coupling specifications

Model	d(mm)	D	L	L1	L2	Torque	V_{max} (r/min)
PC1	8	20	25	8,5	4	1,5	6100



Dimension

Series	ΦD	Φd1、Φd2	L	L1	L2	Clamp Screw	
	mm	mm	mm	Mm	mm	M	Rated Torque (N·m)
PC1	16	3~6	23	7.5	3.5	2.5	1.5
	20	4~8	25	8.5	4	2.5	1.5
	25	5~10	31	10	5	3	2.0
	32	6~12	41	12	6	4	3.0
	40	8~18	56	14	7	5	4.0

Specification

Series	ΦD (mm)	Rated Torque (N·m)	Angular Misalignment	Parallel Misalignment (mm)	Static Tensional Stiffness	Max. Speed (r/min)	Moment Inertia (Kg·m^2)	Axial Motion (mm)
	16	1.0			110 N·m/rad	7600	2.5×10^{-6}	±0.3
	20	2.0			170 N·m/rad	6100	7.1×10^{-6}	±0.4

Figure 3.18 Specifications of the flexible coupling

CHAPTER 4: DESIGN CIRCUITS, PROGRAM CONTROLS AND SOFTWARE FOR LASER MACHINE

4.1 Construct an electrical circuit

4.1.1 Block diagram

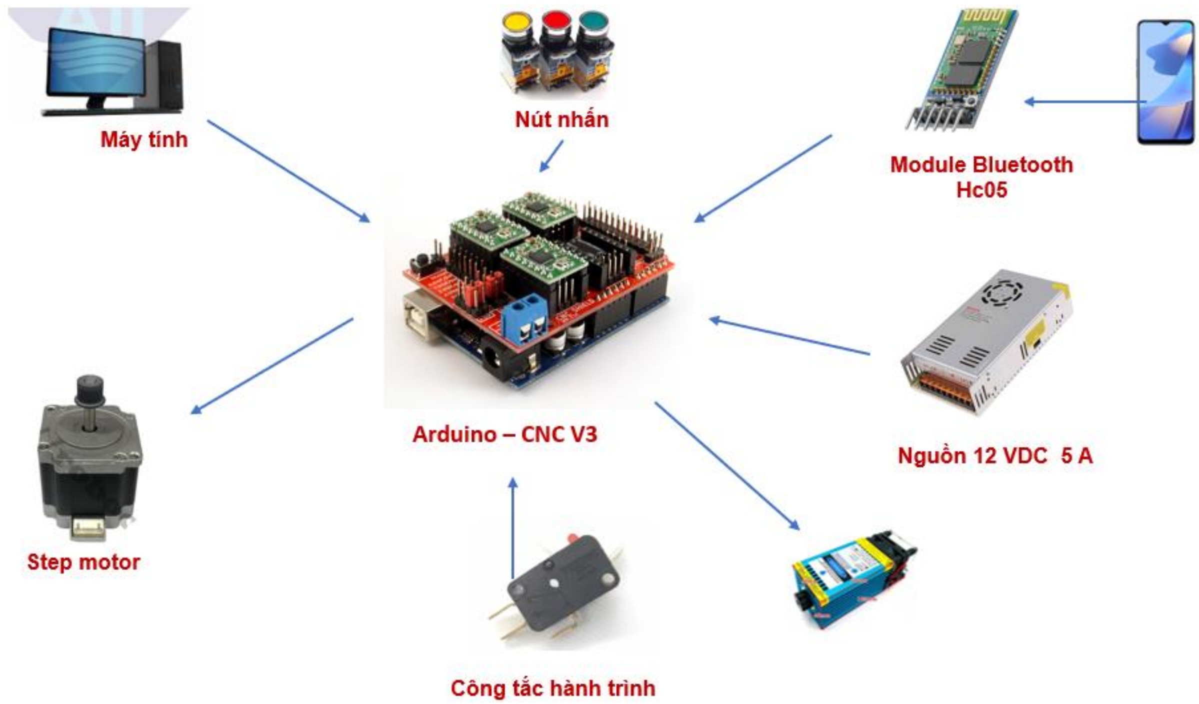


Figure 4.1 System block diagram

+ The power block is used to supply voltage and current to the entire electrical circuit.
+ The computer is used to transmit data through software to automatically control devices via UART communication.
+ The specific microcontroller, Arduino R3, is responsible for controlling and monitoring devices that can communicate with the computer via the Com port.
+ The LCD display block has the function of displaying the interface and coordinates on the 16x2 LCD screen via I2C communication.
+ The push-button block sends input signals from the push-button to the microcontroller for task processing.
+ The motor block receives signals from the microcontroller and is controlled through the A4988 driver.
+ The travel switch sensor block is used to limit the travel of the machine when it runs beyond the operating range.
+ The Laser block receives signals from the microcontroller to adjust the engraving power output suitable for each pixel point.

4.1.2 Electrical circuit diagram

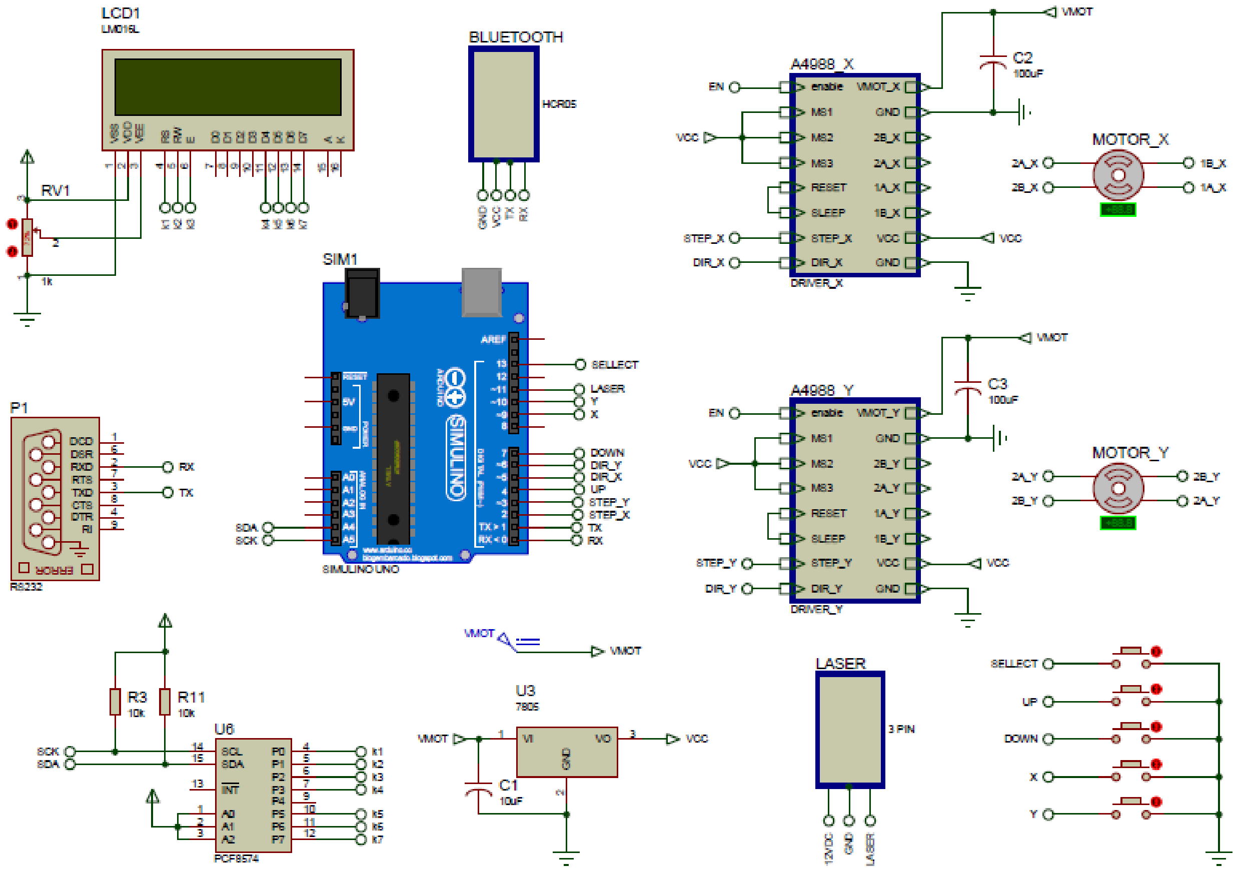


Figure 4.2 Electrical circuit diagram

4.1.3 Electronic components and control system

a) Control circuit Board Arduino Uno R3

- Arduino is now widely known in Vietnam, and in the world it is very popular. Its power is increasingly proven over time with countless unique open source applications that are widely shared. With Arduino you can apply to simple circuits such as light sensor circuit to turn on and off lights, motor control circuit, …

Figure 4.3 Arduino Uno R3

Table 1.7 Some specifications of Arduino UNO R3

Microcontroller	ATmega328 (8-bit family)
Operating voltage	5V - DC (only supplied via USB port)
Operating frequency	16MHz
Current consumption	30mA
Voltage and recommended	7 - 12V - DC
Voltage and recommended	6 - 20 V - DC
Number of Digital I/O pins	14 (6 PWM pins)
Number of Analog pins	6 (10-bit resolution)
Maximum current per I/O pin 30mA	30mA
Maximum output current (5V) 500mA	500mA
Maximum output current (3.3V) 50mA	50mA
Flash memory	32Kb (ATmega328) with 0.5Kb used by bootloader
SRAM	2Kb (ATmega328)

b) Module sheld CNC V3

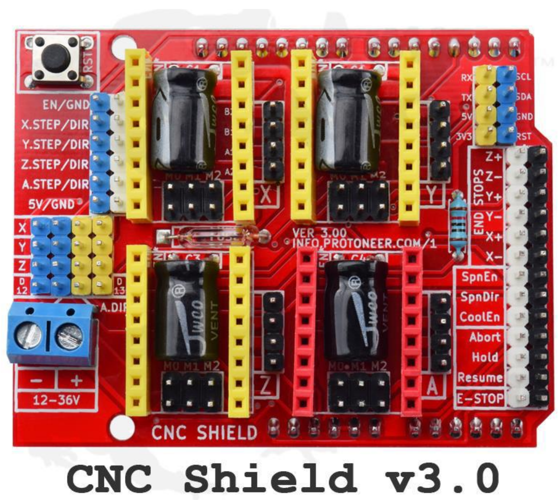


Figure 4.4 CNC Shield V3

Specifications:

- Compatible with Arduino GRBL 0.8c library
- Supports 4 axes, 2 end sensors for each axis
- Connects to Spindle
- Connects to the cooling fan
- Compatible with A4988 / DRV8825 driver
- Has jump set microstep mode for Step driver
- Operating voltage from 12V - 36V

c) Stepper motor and A4988 Driver

- The stepper motor is an electric motor that does not use synchronous switching. Specifically, the poles in the stator and rotor are permanent magnets or in the case of the reluctance motor, they are tooth blocks made from magnetic materials. All reversing circuits must be controlled externally by the controller and especially the motors, the controller is designed so that the motor can hold any fixed position as well as rotate to any position.

Figure 4.5 Step Motor

Stepper motors are divided into 3 basic types:

- Permanent magnet stepper motor
- Variable reluctance stepper motor
- Hybrid stepper motor

Among them, the permanent magnet stepper motor includes 3 types:

+ Unipolar Stepper Motor
+ Bipolar Stepper Motor
+ Hybrid stepping Motor

❖ Unipolar Stepper Motor

+ The unipolar type stepper motor consists of 2 coils, each coil is connected to the outside at the middle of the coil, so usually in practice this is a 5 or 6 wire motor. It is controlled by connecting the common wire head to the source and each remaining wire head is sequentially connected to the mass.

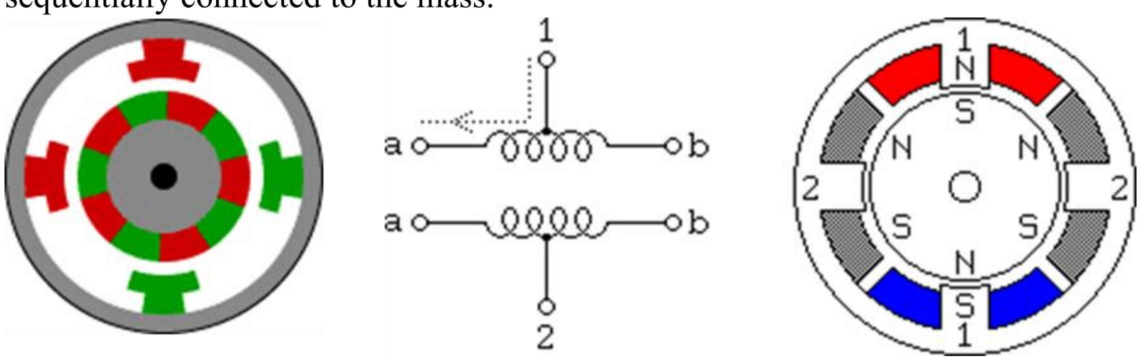


Figure 4.6 Single-pole motor Bipolar Stepper Motor

❖ Bipolar Stepper Motor

+ The bipolar motor has the same structure as the single-pole motor, except that the single-pole motor does not have a common wire between the two wires. The advantage of this type of motor is that the current runs through both coils 1 and 2, creating a large torque, but the motor control circuit is more complicated, we have to reverse the current for the coil.

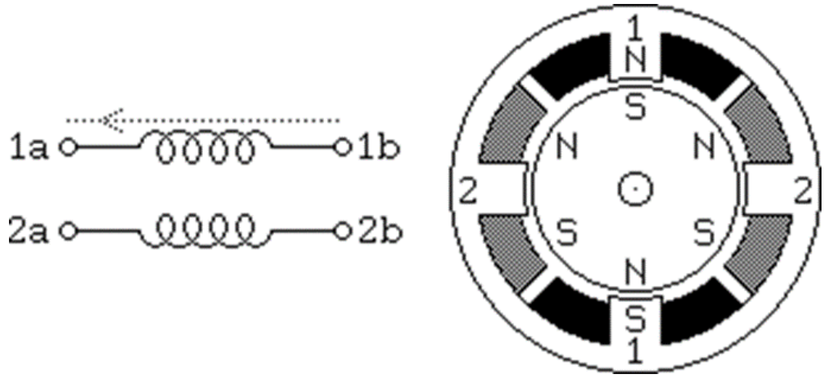


Figure 4.7 Bipolar motor

- A4988 Driver: is a tiny DMOS controller with a converter and overcurrent protection. The A4988 can control a bipolar stepper motor with a current of up to 2A per coil.

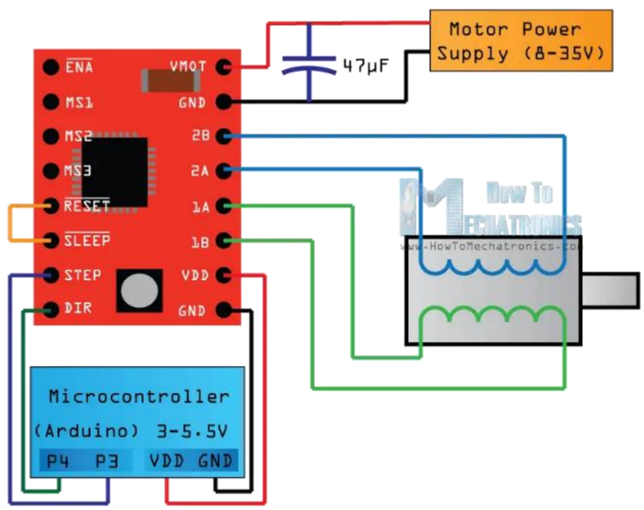


Figure 4.8 Connection diagram of A4988 driver

Below are some main features:

- o Easy to control the direction of rotation and the number of rotations.
- o Control mode: full step, half step, 1/4, 1/8, 1/16
- o The maximum current can be adjusted through a potentiometer allowing the stepper motor to operate at maximum power.
- o The maximum current can be adjusted through a potentiometer allowing the stepper motor to operate at maximum power
- o Short circuit protection.

d) 2.5 W Laser diode

- Based on the financial capability and requirements of the engraving machine on materials such as wood, leather, paper, and carbon, we choose a 2500mW laser head.

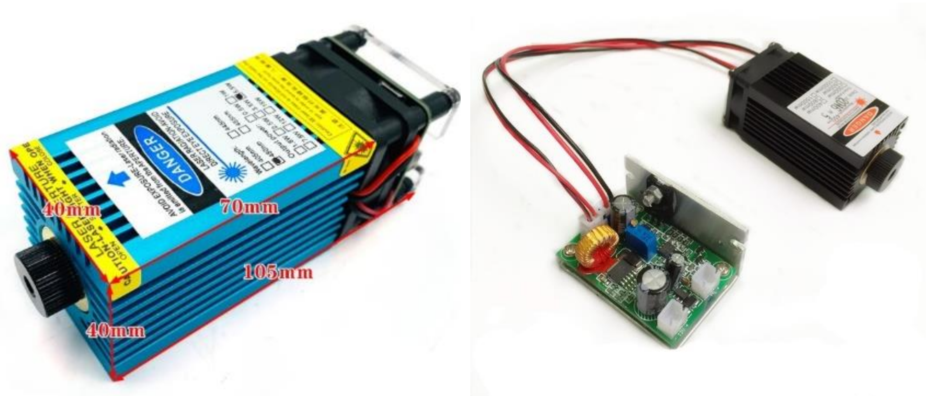


Figure 4.9 Laser diode 2.5W

Specifications:

- Wavelength: 405nm
- Power: 2500mW
- Input voltage: 12V DC – 2A
- Operating temperature: + 10 ℃ ~ + 40
- Cable length: 40cm
- Cable length: 40cm

e) Cable length: 40cm

+ The travel switch is used to limit a certain movement. It consists of one or more NO NC contacts.
+ Travel switches can be normally closed buttons, normally open, 2-contact switches, and even optical switches…
+ When the travel switch is actuated, it will close or break an electrical circuit, so it can interrupt or start another device.

f) LCD 16x2 (I2C)

- The LCD (Liquid Crystal Display) display device is used in many applications of microcontrollers. LCD has many advantages compared to other types of displays: It can display diverse, intuitive characters (letters, numbers, and graphics), easily integrated into application circuits with various communication protocols, consumes very little system resources, and is cheap….

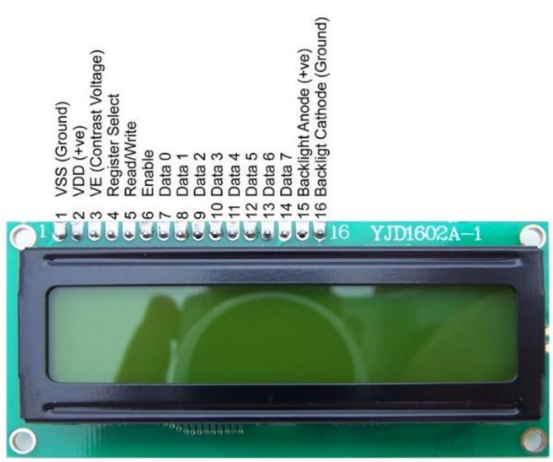


Figure 4.10 Pin diagram of LCD 16x2

- The I2C LCD module is an extension for the regular LCD screen, designed to reduce the number of connection pins through the I2C (Inter-Integrated Circuit) protocol. The I2C protocol helps to transmit data between devices quickly and simply, reducing the complexity of the connection.

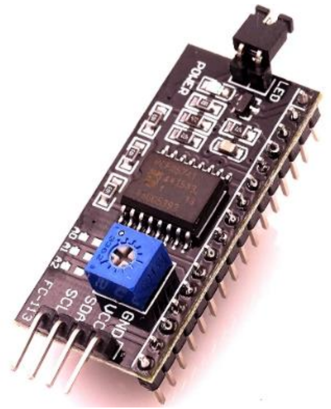


Figure 4.11 Module I2C

❖ Actual circuit diagram and electrical box image

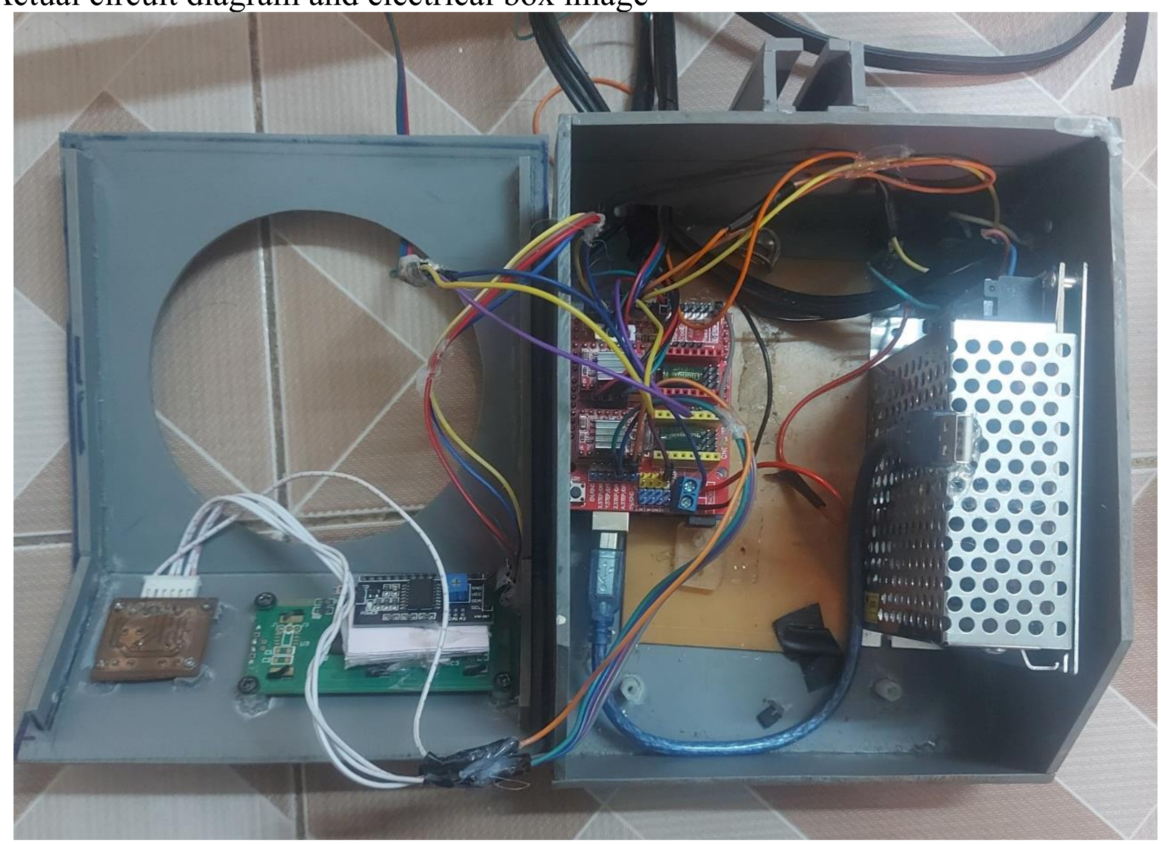


Figure 4.12 Internal electrical connection circuit.

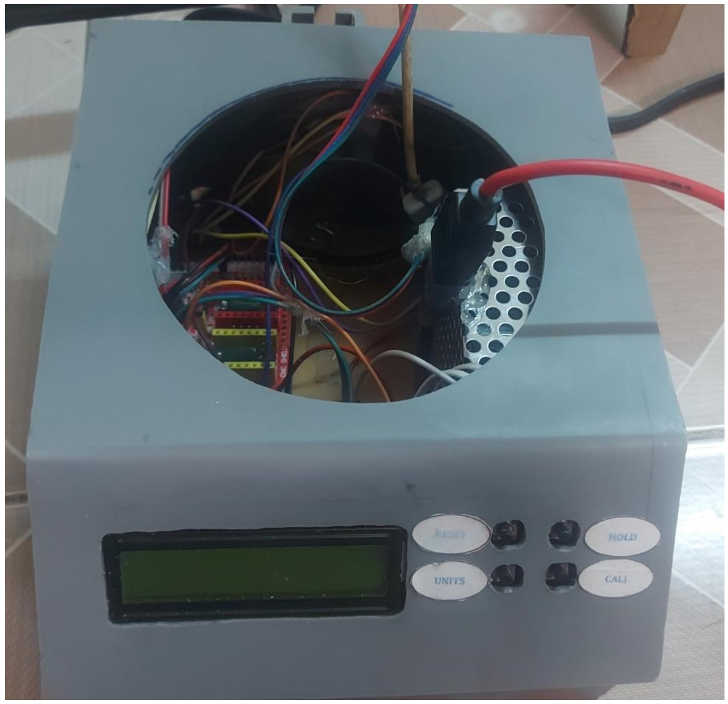


Figure 4.13 Electrical box containing the control circuit.

4.2 Programming for the system

- To evaluate and improve the system, our team decided to use 2 methods:

o Method 1:

Use open-source Firmware + LaserGrbl software for processing.

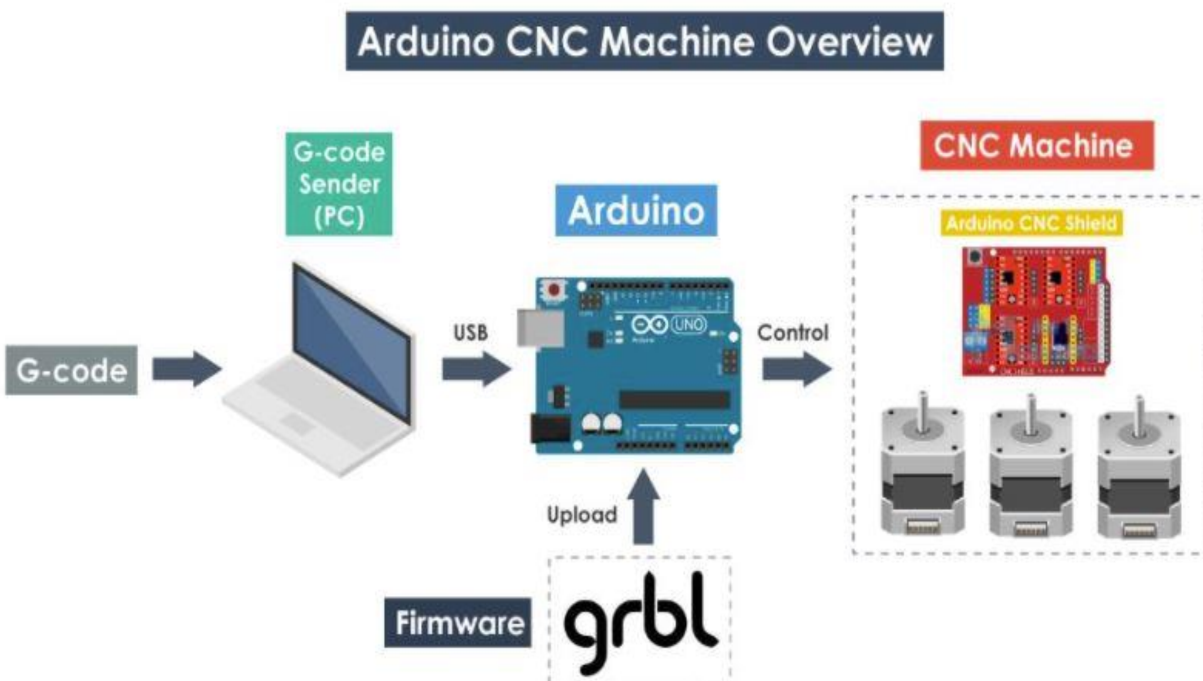


Figure 4.12 Priciple diagram

+ The GRBL library is a high-performance open-source library, it is a solution to replace the commonly used parallel-port-based in CNC milling machines. The GRBL library can operate on most of the current Arduino Classic boards (Arduino UNO, Nano, Pro mini, mini,…). You just need an Arduino board with a storage memory of 30KB or more to make a CNC machine operate immediately.

+ The control library is written in C language optimized to operate with high performance and fully exploit the capabilities of the AVR chip line to achieve accurate timing and multitasking (asynchronous).
+ The GRBL library uses basic G-code scripts and operates accurately on many CNC machine lines without any errors

o Method 2:

Layout Firmware to read Gcode and control the motor + write software on the C# interface

- Purpose: to evaluate, compare the quality, stability, and accuracy of the self-written program compared to the program available on the market that has been optimized and accurate or not.
- To export Gcode for processing, we use LightBurn software to export the code.

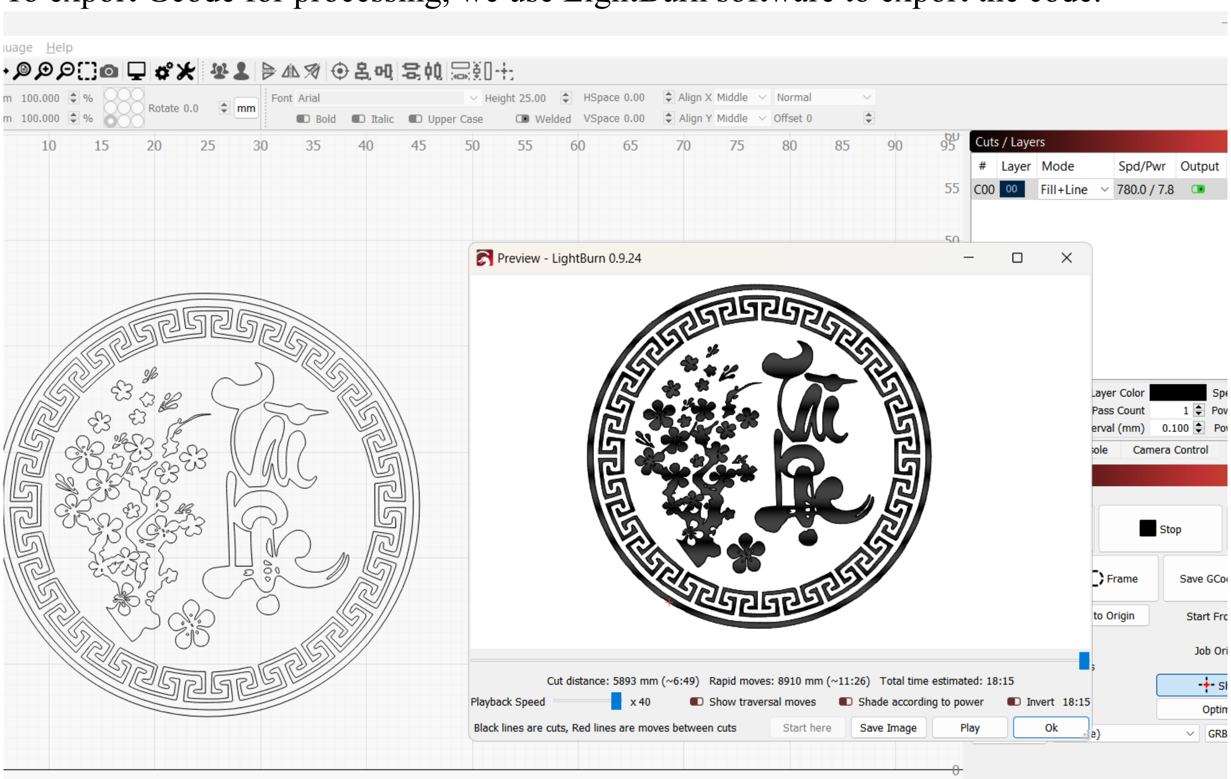


Figure 4.13 Process and creation of Gcode

+ First, we add an image or create it on the software using basic tools
+ Then edit the image and size as required.
+ Next, convert the image to engraving form (line + Fill) to produce an accurate engraving image.
+ Finally, export the Gcode and save it to a file.nc

4.2.1 Flowchart diagram

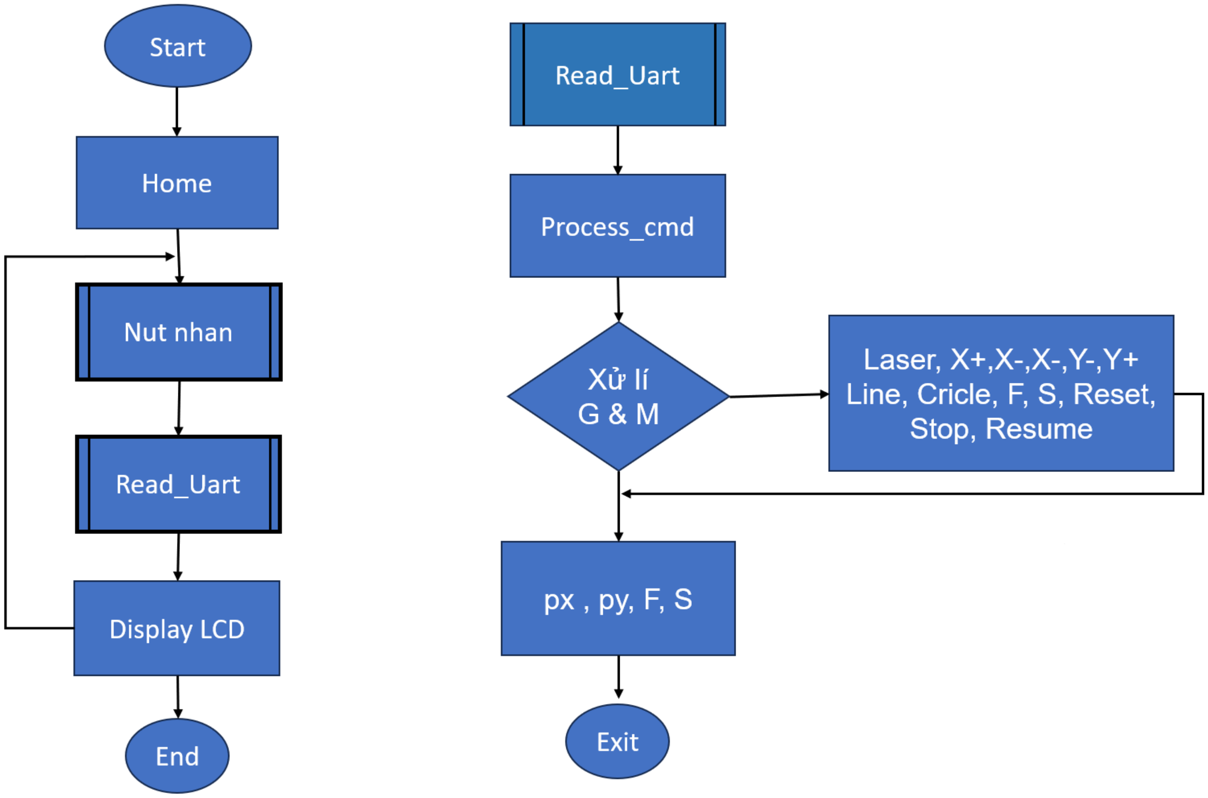


Figure 4.14 Main program

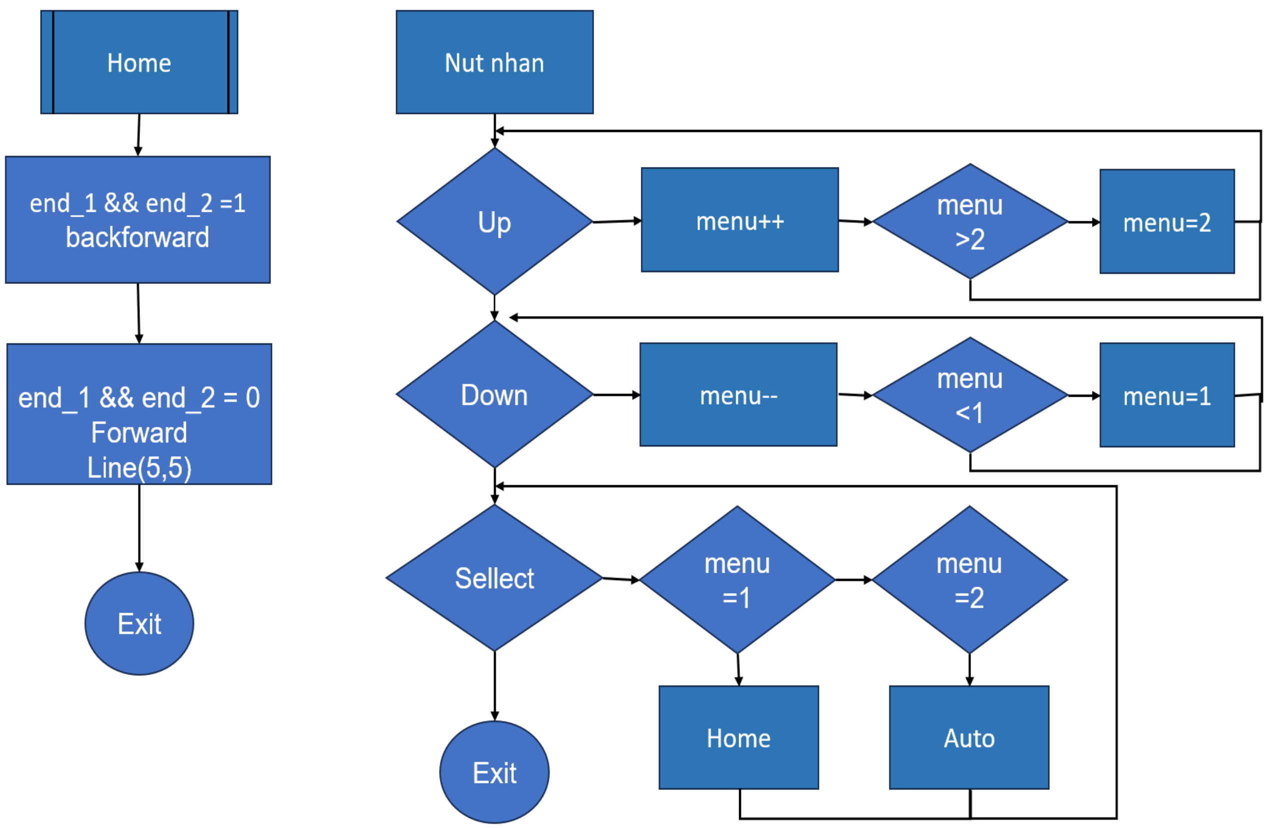


Figure 4.15 Diagram showing LCD tasks using buttons

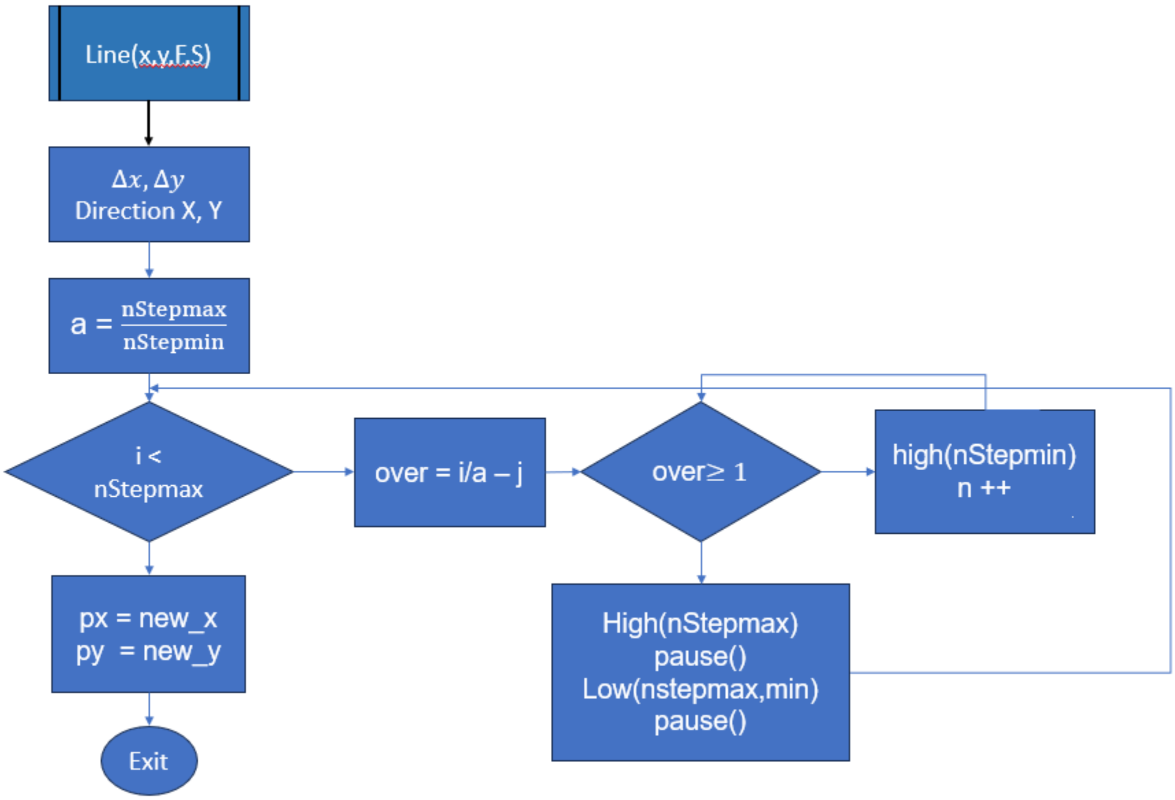


Figure 4.16 Flowchart of the linear interpolation algorithm

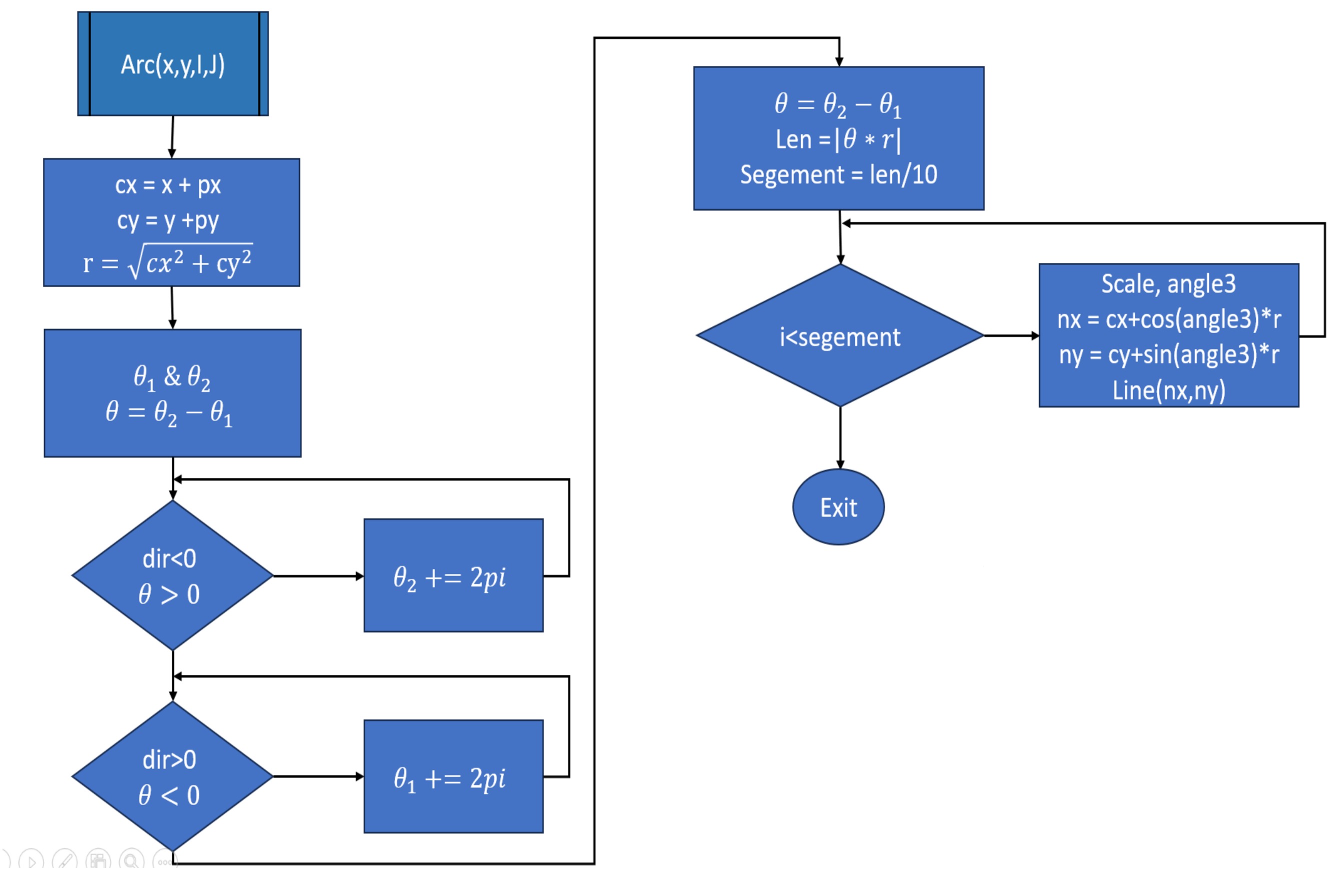


Figure 4.17 Circular arc interpolation

- Gcode command table that the program can handle:

Table 1.8 Processing code

Command	Function
G00	Fast run control
G01	Linear interpolation
G02	Circular interpolation (clockwise)
G03	Circular interpolation (counterclockwise)
G04	Hold time
G20	Inch unit
G21	Millimeter unit
G90	Absolute coordinate system
G91	Relative coordinate system
M5	OFF Laser
M3	ON Laser
M115	Reset Home move(0,0)
M116	X+
M117	X-
M118	Y+
M119	Y-
M120	STOP
M121	RESUME

4.2.2 Operating principle of the system

- The machine has two control modes: manual and automatic control using software written in C#

+ Manual control: On the machine control cabinet, there is a 16x2 LCD display showing the interface, on the left there are 4 buttons including a Reset button used to reset the microcontroller, a Select button used to select functions, an UP button used to press up, a Down button used to press down. When selecting the Home function button, the machine will run back to touch the X and Y travel switches and determine that as the origin and run back a small distance to avoid colliding with the travel switch.
+ Software control: used to control the X and Y axes, load Gcode to run the program, turn on and off the Laser head.

4.3 Software interface design

4.3.1 General introduction about Visual Studio 2022 software

- Visual Studio 2022 software is a fairly new version of Visual Studio software provided by Microsoft. The software can help to program to create interfaces (computer software) in a fairly easy and convenient way. In this topic, the Laser engraving machine control interface will be written using Visual Studio software in C# language.

Figure 4.7 Logo Visual Stuido

4.3.2 Software and user interface

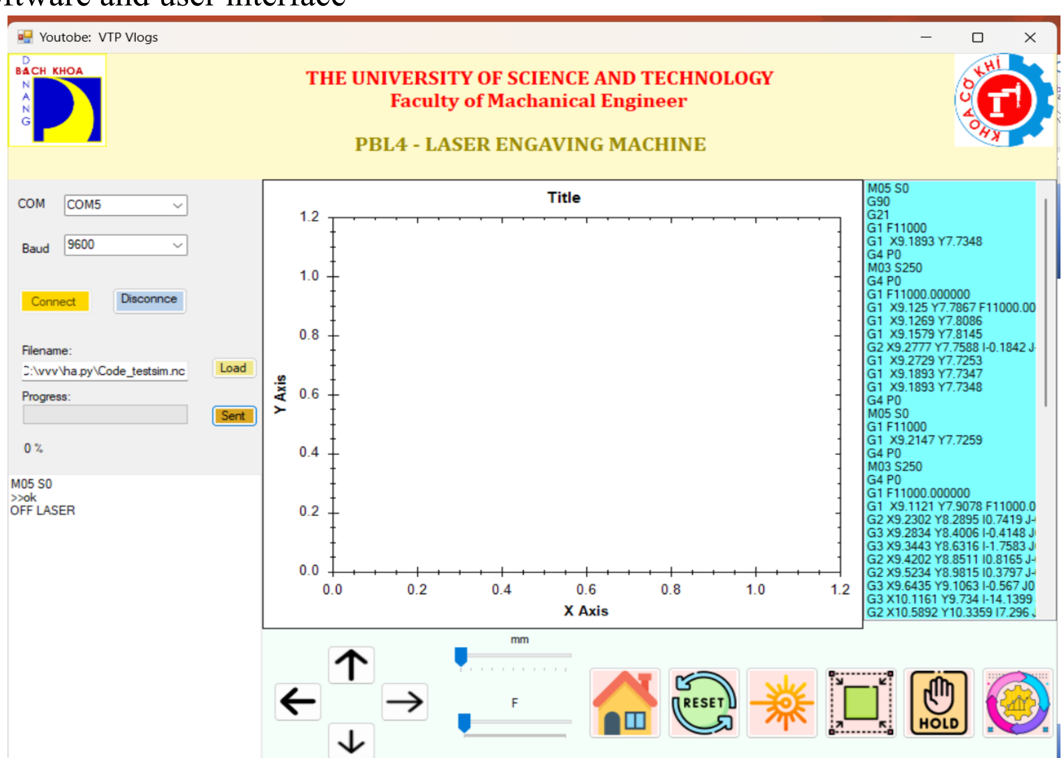


Figure 4.18 Control interface machine self-designed

Advantages:

- Self-design to understand the process
- User interface intuitive and lively

Disadvantages:

- Processing ability is not optimal
- Software does not display real-time position
- Noise due to electrical circuits

To overcome these situations, it is necessary to study deeply about programming and controlling Laser engraving machines to improve and enhance the quality of programming and software.

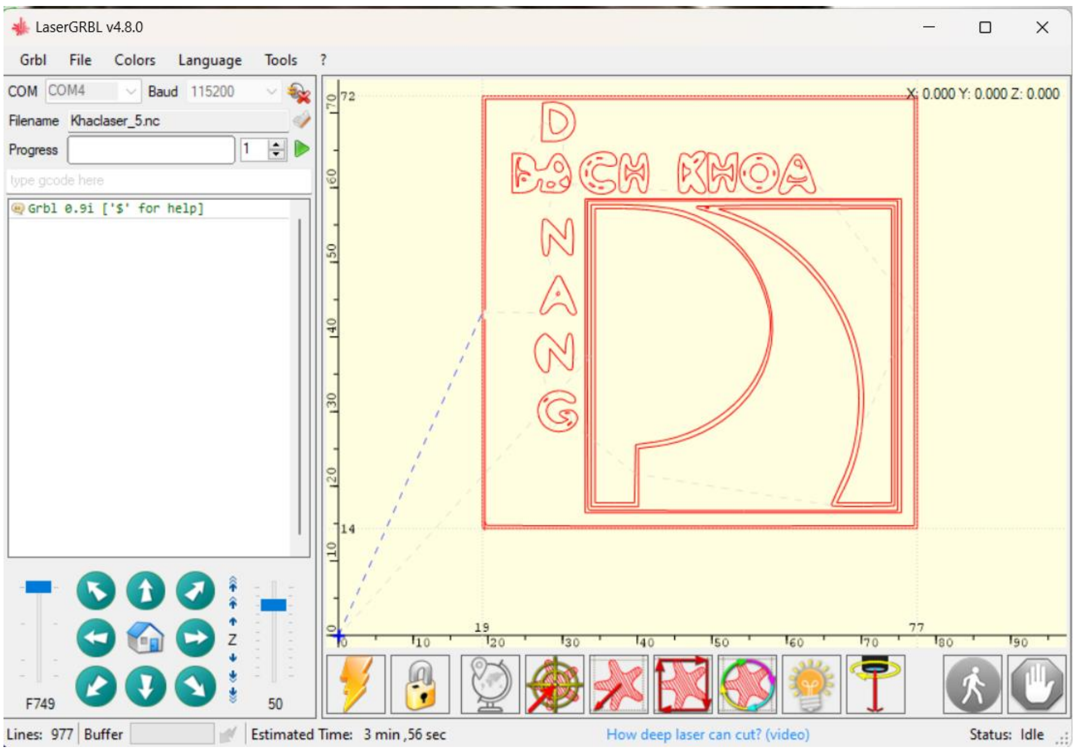


Figure 4.19 LaserGRBL Software

- LaserGRBL is a software that controls laser cutting and engraving machines designed to work with GRBL, an open source software that controls CNC cutting and engraving machines

Advantages:

- Compatible with Grbl Firmware open source code
- Easy to use interface o Simulink simulation feature
- Edit and optimize engraving paths o Support multiple languages
- Send G code directly and have the ability to create Gcode from images
- Send G code directly and have the ability to create Gcode from images

Disadvantages:

- Limited in design and image editing
- Mostly dependent on Firmware
- Limited device connectivity

CHAPTER 5 LESSONS LEARNED, RESULTS AND DEVELOPMENT DIRECTIONS

5.1 Experimental model and results

5.1.1 Experimental model

- The group has successfully built a 2-axis Laser model as shown in Figure 5.1. This model is similar to the original design that was set out.

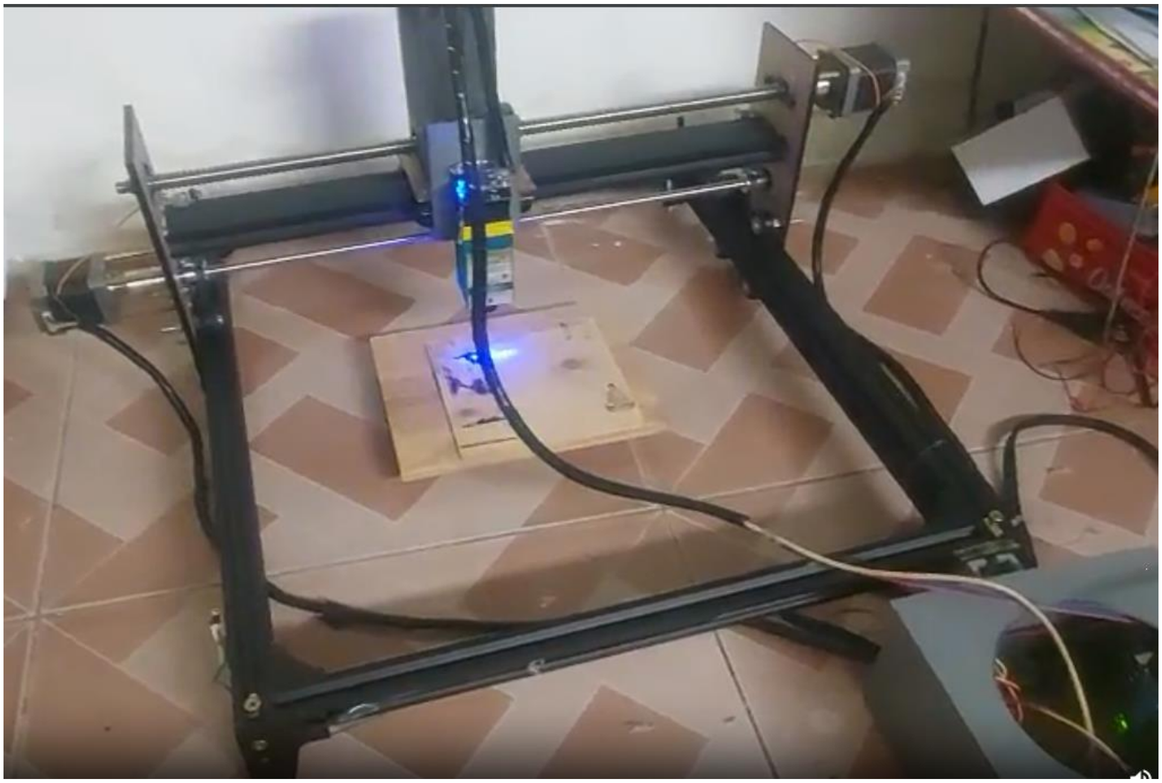


Figure 5.1 Model Laser machine completed

- Technical characteristics of the model are described:

Table 1.9 Parameters and characteristics of the three-axis CNC machine

STT	Parameter	Description
1	Size	+ The model has a total length of 600 mm, a width of about 500 mm, a height of about 300 mm. The model weight is 10 kg.
2	Axis X-Y	+ The X axis of the model uses a belt drive on both sides with a shaft d = 8mm using a GT2 pulley + The Y axis uses a ball screw, placed on two slide bars. + The Z axis uses a ball screw, fixed to the Y axis to adjust the height for the laser head.
3	Operating stroke	Operating stroke: + X axis: 320 mm + Y axis: 350 mm

4	Moving speed of each axis	The moving speed of the X axis is 26 mm/s The moving speed of the Y axis is 26 mm/s.
5	Materials	Aluminum, steel, plastic…
6	Aluminum, steel, plastic…	Error of the machine 0.2 mm Voltage 12V-5A

5.1.2 Result

- The process of construction and laser engraving on wood and carbon materials

Figure 5.2 The result after processing

5.2 Product Evaluation

❖ After testing the product on the model and self-designed software, the following results were achieved:
  + Able to interpolate straight lines and circles
  + Have been able to engrave and cut thin objects like paper, cardboard…

- Comment:
  - The engraving speed is still slow, depending on the material and power of the engraving head
  - The accuracy of the machine is only relatively, not high

- Efficiency is still low.

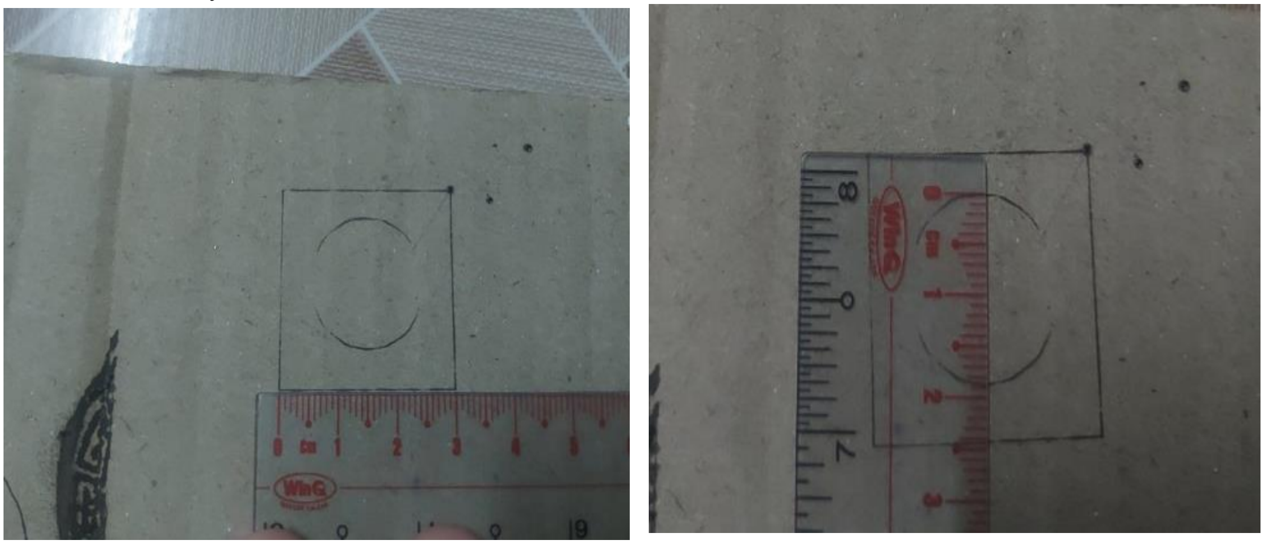


Figure 5.3 The result of running on the self-designed program

5.3 Engraving parameters for each type of material

Table 1.10 Engraving parameters of the machine for each material

Material	Power	Lenght	Velocity	Evaluation
Wood (1)	40/255	52 mm	500 mm/p	Y
Paper	20/255	45 mm	800 mm/p	Y
Carbon	150/255	52 mm	500 mm/p	Y
Wood (2)	50/255	52 mm	750 mm/p	Y
PVC				
Leather				
Xenlulozo				

5.4 Development direction

- Can add features to load Gcode via Wifi or Bluetooth
- Improve the machine's accuracy in interpolation
- Improve the stability of the machine

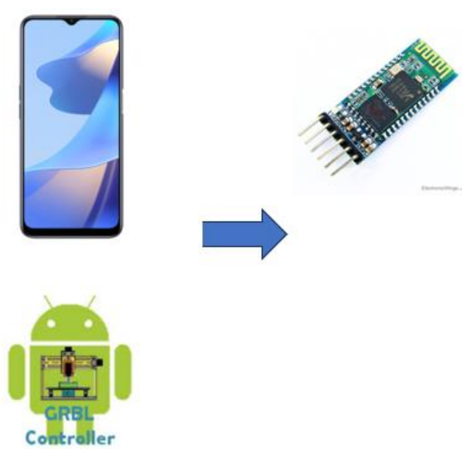


Figure 5.4 Wireless connection

REFERENCE

[1] Nguyễn Trọng Hiệp, Nguyễn Văn Lẫm, "Thiết kế chi tiết máy", NXB Giáo dục, 1999.
[2] Trịnh Chất, Lê Văn Uyển, "Tính toán thiết kế hệ thống dẫn động cơ khí- Tập 1", NXB Giáo dục
[3] Bùi Quý Lực - "Hệ thống điều khiển số trong công nghiệp", NXB Giáo dục
[4] Catalog động cơ bước Nema 17: https://datasheetspdf.com/pdf/1291258/Nidec/KH42KM2-901/1
[5] Silde bài giảng "Bộ truyền đai" - https://cuuduongthancong.com/dlf/747438/chi-tiet-may/phan-tan-tung/ch04_truyen_dong_dai.pdf.
[6] Tính chọn bước với đây đai - https://www.diyeverything.xyz/2020/06/dong-co-buoc-va-cach-tinh-buoc-stepper-motor-nema.html
[7] Catalog tính chọn động cơ bước- Link: Technical Reference Overview | PDF | Transmission (Mechanics) | Torque (scribd.com)

APPENDIX

❖ Code Arduino:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0X27,16,2); //SCL A5 SDA A4

#define LASER          11
#define limit_SW1       9
#define limit_SW2       10
#define upButton       4
#define downButton      7
#define selectButton    13
#define ARC_CW         (1)
#define ARC_CCW        (-1)
#define MM_PER_SEGMENT  (5)
#define STEPS_PER_TURN (200)
#define MIN_STEP_DELAY (0.3)
#define MAX_FEEDRATE   (1800)
#define MIN_FEEDRATE   (0.01)
#define MAX_BUF        (64)

int times = 0;
int power = 0;
int distance = 0;
int laseron = 255;
int laseroff = 0;
int menu = 1;
char  serialBuffer[64];
int   sofar;
float px, py;
float fr = 0;
float  step_delay = 0;
char mode_abs=1;
bool Flag_1 = false;
void m1step(int dir)
{
  digitalWrite(8,LOW);
  digitalWrite(5,dir==1? HIGH:LOW);
}
```

```
void m2step(int dir)
{
  digitalWrite(8,LOW);
  digitalWrite(6,dir==1?HIGH:LOW);
}

void disable()
{
  digitalWrite(8,HIGH);
}

void setup_controller()
{
  pinMode(8,OUTPUT);
  pinMode(2,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(limit_SW1, INPUT);
  pinMode(limit_SW2, INPUT);
  pinMode(upButton, INPUT_PULLUP);
  pinMode(downButton, INPUT_PULLUP);
  pinMode(selectButton, INPUT_PULLUP);
  digitalWrite(8,HIGH);
}

void updateMenu() {
  switch (menu) {
    case 0:
      menu = 1;
      break;
    case 1:
      lcd.clear();
      lcd.print("> HOME");
      lcd.setCursor(0, 1);
      lcd.print(" READ CARD");
      break;
    case 2:
      lcd.clear();
      lcd.print(" HOME");
      lcd.setCursor(0, 1);
```

```
      lcd.print(">READ CRAD");
      break;
    case 3:
      lcd.clear();
      lcd.print(">CONTROL X Y");
      lcd.setCursor(0, 1);
      lcd.print(" LASER ENGAVING");
      break;
    case 4:
      lcd.clear();
      lcd.print("CONTROL X Y");
      lcd.setCursor(0, 1);
      lcd.print(">LASER ENGAVING");
      break;
    case 5:
      menu = 4;
      break;
  }
}
void displayValuesOnLCD(float x, float y, int fr) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("X:");
  lcd.print(x);  // In giá trị x lên LCD

  lcd.setCursor(9, 0);
  lcd.print("Y:");
  lcd.print(y);  // In giá trị y lên LCD

  lcd.setCursor(0, 1);
  lcd.println("F:");
  lcd.setCursor(2, 1);
  lcd.print(int(fr));  // In giá trị y lên LCD
  lcd.print("        ");
}
  // Hien thi ra man hinh
void ready() {
  sofar=0;  // clear input buffer
  Serial.print(F(">>"));
}
```

```
void Send_data(){
   ready();
   Serial.println("ok");
}

void pause(long ms)
{
  delay(ms);
}
void feedrate(float nfr)
{
 if(fr==nfr) return;
 if(nfr>MAX_FEEDRATE || nfr<MIN_FEEDRATE)
{
    Serial.print(F("New feedrate must be greater than "));
    Serial.print(MIN_FEEDRATE);
    Serial.print(F("steps/s and less than "));
    Serial.print(MAX_FEEDRATE);
    Serial.println(F("steps/s."));
    return;
  }
  step_delay = 1000/nfr;
  Serial.println(step_delay);
  if (step_delay < MIN_STEP_DELAY){
    step_delay = MIN_STEP_DELAY;
    Serial.println(" max roi");
  }
  fr = nfr;
}
void LaserPower (int mypower){
  if (mypower == laseron)
  {
      analogWrite(LASER,power);
  }
  else if(mypower == laseroff)
  {
      analogWrite(LASER,power);
  }
}

void position(float npx,float npy)
```

```
{
  px=npx;
  py=npy;

}
void line(float newx,float newy)
{
  if (newx == px && newy == py)
  {
    return;
  }
  float over= 0;
  float steps_axis_min = 0;
  float a =0 ;
  float dx  = newx-px;
  float dy  = newy-py;
  int dirx = dx>0?1:-1;
  int diry = dy>0?1:-1;
  float nStepMax = 0;
  float nStepMin = 0;
  int stepPinMin = 0;
  int stepPinMax = 0;
  dx = abs(dx);
  dy = abs(dy);
  if((float)dx*80 > (float)dy*800) {
    nStepMax = dx*80;
    nStepMin = dy*800;
    stepPinMax = 2;
    stepPinMin = 3;
  }
  else
  {
    nStepMax = dy*800;
    nStepMin = dx*80;
    stepPinMax = 3;
    stepPinMin = 2;
  }
  digitalWrite(5,dirx==1? HIGH:LOW);
  digitalWrite(6,diry==1? HIGH:LOW);
  digitalWrite(8,LOW);
  if(nStepMin != 0){
```

```
  a = nStepMax/nStepMin;
  for(float i=1; i<= nStepMax;i=i+1)
  {
    over = i/(a);
    if(over - steps_axis_min >=1)
    {
      digitalWrite(stepPinMin, HIGH);
      steps_axis_min++;
    }
    digitalWrite(stepPinMax, HIGH);
    pause(step_delay);
    digitalWrite(stepPinMin, LOW);
    digitalWrite(stepPinMax, LOW);
    pause(step_delay);
  }
}
else{
    if (dx == 0) {
      for(float i=1; i<= nStepMax;i=i+1) {
        digitalWrite(stepPinMax, HIGH);
        pause(step_delay);
        digitalWrite(stepPinMax, LOW);
        pause(step_delay);
      }
    }
   else if(dy == 0) {
    // Only movement along the x-axis
      for(float i=1; i<= nStepMax;i=i+1) {
        digitalWrite(stepPinMax, HIGH);
        pause(step_delay*10);
        digitalWrite(stepPinMax, LOW);
        pause(step_delay*10);
      }
  }
}
Serial.println(a);
Serial.println(nStepMax);
px = newx;
py = newy;
displayValuesOnLCD(px, py, fr);
digitalWrite(8,HIGH);
```

```
}

float BBatan(float thisdx,float thisdy,float i, float j)
{
  float a = atan2(thisdy-j,thisdx-i);
  if(a<0) a = (PI*2.0)+a;
  return a;
}

void swap(double* k, double* m)
{
    double temp = *k;
    *k = *m;
    *m = temp;
}
void arc(float cx,float cy,float x,float y,int dir)
{
  float point1 = px + cx;
  float point2 = py + cy;
  float radius=sqrt((cx*cx)+(cy*cy));
  double theta1 =BBatan(point1,point2,px,py);
  double theta2 =BBatan(point1,point2,x,y);
  double delta_theta ;
  if (dir <0) {
    swap(&theta1, &theta2);
  }
  delta_theta = theta2-theta1;
  if (delta_theta < 0)
  {
      delta_theta += 2 * PI;
  }
  double len = abs(delta_theta * radius);
  int i, segments = ceil( len / MM_PER_SEGMENT );
  float nx, ny, angle3, scale;
  for(i=0;i<segments;++i)
  {
    scale = ((float)i)/((float)segments);
    angle3 = ( delta_theta * scale ) + theta1;
    nx = point1 + cos(angle3) * radius;
    ny = point2 + sin(angle3) * radius;
    line(nx,ny);
```

```
 }
 Serial.println("drwing final line");
 displayValuesOnLCD(px, py, fr);
}
float parseNumber(char code,float val) {
 char *ptr=serialBuffer;
 while((long)ptr > 1 && (*ptr) && (long)ptr < (long)serialBuffer+sofar)
 {
  if(*ptr==code)
  {
   return atof(ptr+1);
  }
  ptr=ptr+1;
 }
 return val;
}
void output(const char *code,float val)
{
 Serial.print(code);
 Serial.println(val);
}

void Homing(){
 int end_1 = 0;
 int end_2 = 0;
 float i = 1;
 float axits_min = 0;
 feedrate(1200);
 displayValuesOnLCD(px, py, fr);
 m1step(-1);
 m2step(-1);
 while(end_1 == 0 || end_2 == 0){
  i++;
  if(end_2 == 0){
   digitalWrite(3, HIGH);
   if(digitalRead(limit_SW2)){
    end_2 = 1;
    Serial.println("ok");
   }
  }
  if((i/10 - axits_min) >= 1 && end_1 == 0) {
```

```
      digitalWrite(2, HIGH);
      axits_min++;
      if(digitalRead(limit_SW1)){
        end_1 = 1;
        Serial.println("limit 1");
      }
    }
    pause(step_delay);
    digitalWrite(2, LOW);
    digitalWrite(3,LOW);
    pause(step_delay);
  }
  while(end_1 == 1 && end_2 == 1){
    px = 0;
    py = 0;
    m1step(1);
    m2step(1);
    line(5,5);
    break;
  }
  px = 0;
  py = 0;
  Serial.println("Homing Complete");
  digitalWrite(8,HIGH);
}
void action1() {
  lcd.clear();
  // Su kien 1
  //lcd.print(">Executing #1");
  displayValuesOnLCD(px, py, fr);
  Homing();
  delay(1500);
}
void action2() {
  lcd.clear();
  lcd.print(">Executing #2");
  delay(1500);
}
void action3() {
  lcd.clear();
  lcd.print(">Executing #3");
```

```
  delay(1500);
}
void action4() {
  lcd.clear();
  Flag_1 = true;

}
void executeAction() {
  switch (menu) {
    case 1:
      action1();
      break;
    case 2:
      action2();
      break;
    case 3:
      action3();
      break;
    case 4:
      action4();
      break;
  }
}

void monitorbuttons(){                    // Check nut nhan
  if (!digitalRead(downButton)){
    menu++;
    updateMenu();
    delay(100);
    while (!digitalRead(downButton));
  }
  if (!digitalRead(upButton)){
    menu--;
    updateMenu();
    delay(100);
    while(!digitalRead(upButton));
  }
  if (!digitalRead(selectButton)){
    executeAction();
    updateMenu();
    delay(100);
```

```
    while (!digitalRead(selectButton));
  }

}
  //print the current position, feedrate, and absolute mode.
void where() {
  output("X",px);
  output("Y",py);
  output("F",fr);
  Serial.println(mode_abs?"ABS":"REL");
}
void processCommand() {
  int cmd = parseNumber('G',-1);
  switch(cmd) {
  case  0:
  case  1: { // line
    feedrate(parseNumber('F',fr));
    power = (int)parseNumber('S',power);
    analogWrite(LASER,power);
    line( parseNumber('X',(mode_abs?px:0)) + (mode_abs?0:px),
          parseNumber('Y',(mode_abs?py:0)) + (mode_abs?0:py) );
    Send_data();
    break;
    }
  case 2:
  case 3: {  // arc
      feedrate(parseNumber('F',fr));
      arc(parseNumber('I',(mode_abs?px:0)) + (mode_abs?0:px),
          parseNumber('J',(mode_abs?py:0)) + (mode_abs?0:py),
          parseNumber('X',(mode_abs?px:0)) + (mode_abs?0:px),
          parseNumber('Y',(mode_abs?py:0)) + (mode_abs?0:py),
          (cmd==2) ? -1 : 1);
      Send_data();
      break;
    }
  case  4:  pause(parseNumber('P',0)*1000);
    Send_data();
    break;  // dwell
  case 21:
    Send_data();
    break;
```

```
case 90:  mode_abs=1;
  Send_data();
  break;
case 91:  mode_abs=0;
  Send_data();
  break;
case 92:
  position( parseNumber('X',0),parseNumber('Y',0) );
  Send_data();
  break;
default:  break;
}

int cmd1 = parseNumber('M',-1);
switch(cmd1) {
case 3:
    power = (int)parseNumber('S',power);
    LaserPower(laseron);
    Send_data();
    Serial.println("ON LASER");
    break;
case 5:
    power = 0;
    LaserPower(laseroff);
    Send_data();
    Serial.println("OFF LASER");
    break;
case 18
  disable();
  Send_data();
  break;
case 114:  where();
  Send_data();
  break;
case 115: Homing();
    feedrate(parseNumber('F',fr));
    displayValuesOnLCD(px, py, fr);
    Serial.println("Home ne");
    sofar=0;  // clear input buffer
    break;
case 116:
```

```
    distance = (int)parseNumber('m',distance);
    feedrate(parseNumber('F',fr));
    line(px+distance,py);
    Serial.println("Run X+");
    sofar=0;  // clear input buffer
    break;
case 117:
    distance = (int)parseNumber('m',distance);
    feedrate(parseNumber('F',fr));
    line(px,py+distance);
    Serial.println("Run Y+");
    sofar=0;  // clear input buffer
    break;
case 118:
    distance = (int)parseNumber('m',distance);
    feedrate(parseNumber('F',fr));
    line(px-distance,py);
    Serial.println("Run X-");
    sofar=0;  // clear input buffer
    break;
case 119:
    //feedrate(1000);
    distance = (int)parseNumber('m',distance);
    feedrate(parseNumber('F',fr));
    line(px,py-distance);
    Serial.println("Run Y-");
    sofar=0;  // clear input buffer
    break;
case 120:
    px =0; py = 0;                    // reset px va py
    displayValuesOnLCD(px, py, fr);
    Serial.println("Reset X Y");
    sofar=0;  // clear input buffer
    break;
case 121:
    Serial.println("ok");
    sofar=0;  // clear input buffer
    break;
default:  break;
}
//Serial.println("Thoat2");
```

```
}
void monitorserial(){
  while(Serial.available() > 0) {  // if something is available
    char c=Serial.read();  // get it
    Serial.print(c);  // repeat it back so I know you got the message
    if(sofar<MAX_BUF-1) serialBuffer[sofar++]=c;  // store it
    if((c=='\n') || (c == '\r')) {
      serialBuffer[sofar]=0;  // end the buffer so string functions work right
      Serial.print(F("\r\n"));  // echo a return character for humans
      processCommand();  // do something with the command
      //Serial.println("thoat1");
    }
  }
}
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  setup_controller();
  updateMenu();
  position(0,0);
  ready();
  Serial.println("Firmware connect");
  LaserPower(laseroff);
  action1();
  updateMenu();
}
void loop() {
    if(Flag_1 == true){
      monitorserial();
      if (!digitalRead(selectButton)){
        Flag_1 = false;
        menu = 4;
        delay(100);
        while (!digitalRead(selectButton));
        lcd.clear();
        lcd.print("CONTROL X Y");
        lcd.setCursor(0, 1);
        lcd.print(">LASER ENGAVING");
      }
      if (!digitalRead(upButton)){
```

```
      displayValuesOnLCD(px, py, fr);
      delay(100);
      while (!digitalRead(upButton));
    }
  }
  else{
    monitorbuttons();
  }
}
```

❖ Code C# Phần mềm

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Text;
using System.Windows.Forms;
using System.Xml;
using System.IO.Ports;
using ZedGraph;
using System.Linq;
using System.Drawing;

namespace PBL4_Software_Laser
{
    public partial class frmMain : Form
    {
        public frmMain()
        {
            InitializeComponent();
            btnConnect.Enabled = true;
            btnDisconnect.Enabled = false;
            btnUp.Enabled = false;
            btnDown.Enabled = false;
            btnRight.Enabled = false;
            btnLaser.Enabled = false;
            btnLeft.Enabled = false;
            btnHome.Enabled = false;
            btnCenter.Enabled = false;
            btnReset.Enabled = false;
            btnLoad.Enabled = false;
            btnSent.Enabled = false;
```

```
        btnResume.Enabled = false;
        btnHold.Enabled = false;

    }
    private List<string> linesToSend = new List<string>();              // Danh sách các dòng
cần gửi
    private bool isSending = false;                                     // Cờ để kiểm tra xem đang trong
quá trình gửi hay không
    private int laserButtonClickCount = 0;
    public bool Hold = false;
    private void btnConnect_Click(object sender, EventArgs e)
    {
        serialPort1.PortName = cbxCom.Text.Trim();
        serialPort1.BaudRate = Convert.ToInt32(cbxBaud.Text);
        serialPort1.Open();
        btnConnect.Enabled = false;
        btnDisconnect.Enabled = true;
        btnResume.Enabled = true;
        btnHold.Enabled = true;
        btnUp.Enabled = true;
        btnDown.Enabled = true;
        btnRight.Enabled = true;
        btnLaser.Enabled = true;
        btnLeft.Enabled = true;
        btnHome.Enabled = true;
        btnCenter.Enabled = true;
        btnReset.Enabled = true;
        btnLoad.Enabled = true;
        btnSent.Enabled = true;
        timer1.Start();
        MessageBox.Show("Kết nối thành công");
    }

    private void btnDisconnect_Click(object sender, EventArgs e)
    {
       serialPort1.Write("M5S0\n");
        serialPort1.Close();
        btnConnect.Enabled = true;
        btnDisconnect.Enabled = false;
        btnResume.Enabled = false;
        btnHold.Enabled = false;
```

```
    btnUp.Enabled = false;
    btnDown.Enabled = false;
    btnRight.Enabled = false;
    btnLaser.Enabled = false;
    btnLeft.Enabled = false;
    btnHome.Enabled = false;
    btnCenter.Enabled = false;
    btnReset.Enabled = false;
    btnLoad.Enabled = false;
    btnSent.Enabled = false;
    timer1.Stop();
}

private void ProcessReceivedData(string data)
{

    if (data.Contains("ok") && isSending)
    {
        SendNextLine();
    }
}
private void AppendDataToListBox(string data)
{
    if (lstGcode.InvokeRequired)
    {
        lstGcode.Invoke(new Action(() => AppendDataToListBox(data)));
    }
    else
    {
        lstGcode.Items.Add(data);
    }
}
private void SendNextLine()
{
    if (serialPort1.IsOpen && linesToSend.Count > 0 && !Hold)
    {
        string line = linesToSend[0];
        linesToSend.RemoveAt(0);
        serialPort1.WriteLine(line);
    }
    else
```

```
    {
        isSending = false;
    }

}

private void StartSending()
{
    linesToSend.Clear();
    linesToSend.AddRange(lstGcode.Items.Cast<string>());
    isSending = true; // Bắt đầu gửi dòng đầu tiên
    SendNextLine();
}

private void btnLoad_Click(object sender, EventArgs e)
{
    OpenFileDialog open = new OpenFileDialog();
    open.Filter = "Tệp NC (*.nc)|*.nc";

    if (open.ShowDialog() == DialogResult.OK)
    {
        txtFilename.Text = open.FileName;
        try
        {
            // Mở tệp và đọc nội dung
            using (StreamReader reader = new StreamReader(open.FileName))
            {
                // Xóa nội dung hiện tại của ListBox
                lstGcode.Items.Clear();

                string line;
                while ((line = reader.ReadLine()) != null)
                {
                    // Thêm từng dòng vào ListBox
                    lstGcode.Items.Add(line);
                }
                reader.Close();
            }
        }
        catch (Exception ex)
        {
```

```
                MessageBox.Show("Lỗi: " + ex.Message);
            }

        }
    }
    private void btnSent_Click(object sender, EventArgs e)
    {
        StartSending();

    }

    private void btnUp_Click(object sender, EventArgs e)
    {
        float a = float.Parse(txtmm.Text);
        float b = float.Parse(txtF.Text);
        string command = String.Format("M117 m{0} F{1}\n", a, b);
        serialPort1.Write(command);

    }

    private void btnRight_Click(object sender, EventArgs e)
    {
        float a = float.Parse(txtmm.Text);
        float b = float.Parse(txtF.Text);
        string command = String.Format("M116 m{0} F{1}\n", a, b);
        serialPort1.Write(command);
        txtNotify.Text = command +"\n";

    }

    private void btnDown_Click(object sender, EventArgs e)
    {
        float a = float.Parse(txtmm.Text);
        float b = float.Parse(txtF.Text);
        string command = String.Format("M119 m{0} F{1}\n", a, b);
        serialPort1.Write(command);
    }

    private void btnLeft_Click(object sender, EventArgs e)
    {
        float a = float.Parse(txtmm.Text);
```

```
    float b = float.Parse(txtF.Text);
    string command = String.Format("M118 m{0} F{1}\n", a, b);
    serialPort1.Write(command);
}

private void btnHome_Click(object sender, EventArgs e)
{
    float b = float.Parse(txtF.Text);
    string command = String.Format("M115 F{0}\n",b);
    serialPort1.Write(command);
}

private void btnReset_Click(object sender, EventArgs e)
{
    //serialPort1.Write("M120\n");
    txtNotify.Clear();

}

private void btnLaser_Click(object sender, EventArgs e)
{
    if (laserButtonClickCount % 2 == 0)
    {
        // Even click count, change color to yellow and send "M3 S255"
        btnLaser.BackColor = Color.Orange;
        serialPort1.Write("M3 S10\n");
    }
    else
    {
        btnLaser.BackColor = SystemColors.Control;
        serialPort1.Write("M5S0\n");
    }

    laserButtonClickCount++;
}

private void btnCenter_Click(object sender, EventArgs e)
{
    // update sau
}
```

```
        private void btnHold_Click(object sender, EventArgs e)
        {;
            btnHold.BackColor = Color.Red;
            btnResume.BackColor = SystemColors.Control;
            Hold = true;

        }
        private void btnResume_Click(object sender, EventArgs e)
        {
            btnHold.BackColor = SystemColors.Control;
            btnResume.BackColor = Color.Yellow;
            Hold= false;
            serialPort1.Write("M121\n");
            isSending = true;
            SendNextLine();

        }
        private void timer1_Tick(object sender, EventArgs e)
        {
            string data = serialPort1.ReadExisting();
            BeginInvoke(new Action(() =>
            {
                if (data != "")
                    txtNotify.Text = data;
                ProcessReceivedData(data); // Xử lý dữ liệu nhận được ở đây
            }));

        }
        private void trackBar1_Scroll(object sender, EventArgs e)
        {
            txtmm.Text = trackBar1.Value.ToString();
        }

        private void trackBar2_Scroll(object sender, EventArgs e)
        {
            txtF.Text = trackBar2.Value.ToString();
        }
    }
}
```