CHAPTER 1
INTRODUCTION

1.1 Background:-

The world has become a high technology with a lot of things becoming smaller and thinner. The fast-growing development of technology and manufacturing, Industrial requirement such as good and high precision quality has helped in developing the CNC machine plotter all of those can be achieved through machines that can be controlled by computers such as Computer Numerical Control (CNC) machine. To implement CNC plotter machine, several concepts must be understood such as: understanding fundamentals, Machine Mechanical design, CNC machine hardware, software developing, test each one of three axis stepper motors and connecting CNC Machine with the software tools and test it, the steps that has been used to implement this project. Three axes of CNC plotter machine can do movement starting with three primary axes which are X, Y and Z axis. The Z axis is being paralleled with the X-axis

Education system plays a major role in describing the innovative ideas to the students. In recent years, the definition of a robot is generally used to mean an unmanned system or automation, as often seen in industrial applications, deep sea planetary probes. Historically speaking, a robot used to be shaped like humans, and referred to as machines and electric systems that were capable of performing similar actions as humans. In this highly developing society; time and man power are critical constrains for completion of task in large scales. The automation is playing important role to save human efforts in most of the regular and frequently carried works e.g. most of the industrial jobs like welding, painting, assembly, container filling etc.

The present open record for every one of the exchanges and that at any point occurred in the system, the development is steady and the extent of the system likewise develops in parallel. The record is ethical and can without much of a stretch convey on the arrangement of substances of the entire framework surrendered to the lack of interest. Homework composing machine is an auto composing machine through which you can make your work simple by programing your

Venture. According to the title this is a straightforward task utilizing Arduino to make

Homework composing machine at your home. This machine can draw any outline and compose any sort of fonts. You can see sharpness and flawlessness of writing in photographs. The machine utilizes a gantry to move the composition tip along the X on an and Y tomahawks. The flexible-nib pen is mounted servo engine which turns the tip onto the composition surface, dealing with the third hub.

In the present study, the X-Y plotter is designed to recording and plotting two-dimensional data on a rectangular coordinate system the material selection of the mechanism was made considering the cost and wide range of applications such as servo motor. Servo motor can be differentiated through the cost, peak torque capability, speed range to compromise the standard and application of the system Park, discuss the dynamics of a dual-drive servo mechanism and develops an XY gantry model consisting of two motors for Y control with another motor sliding the gantry in the X direction. The design uses two parallel rails for Y-motion with a bar spanning across the rails which holds the end effectors of the system. In the other hand, the accuracy of plotting is the main issue to be concerned on the fabrication of X-Y plotter. Few papers have been devoted to plotters in term of their adjustment methods on the accuracy and movement of plotter The X-Y plotter system is a more simplified system comparing to the CNC system since CNC system is running on 3 axis direction and the programming is more complicated Nevertheless, the coding for the CNC system which is the G-Code programming can be modified and simplify it to the 2 axis coding movement. G-Code is the generic name for a control language for Reprap machines.

It is a function to tell the machine to move to various points at the desired speed, control the spindle speed, and turn on and off various coolants. In this X-Y plotter system, G-Code is employed by the part programmer to specifying the coordinates of the point to be moved and giving the normal vector to the surface at that point. For the core system, Arduino system is most familiar by the inventor and mainly used in most of the electronic components because of is compatibility of the system with the hardware. Meanwhile, low cost and easily controlled function of the Arduino system contributed on simplifying the building circuit of the microcontroller in the X-Y plotter.

1.2 Objectives:-

- The objectives of this project are to design the CNC Plotter Machine and to develop open source software and hardware to control it.It’s making easier to do the homework or wooden and acrylic carving project.
- The machine create accurate output Low cost CNC machines are perfect for offering preparing students especially in the field of Graduation The littler size make it compact, portable and the segments can effortlessly be dismantled or gather
- . The size, space and vitality required for the machine is additionally lessened. It presently requires just less material and parts to make the machine, consequently cutting down the cost extraordinarily This proposal outlines the planned construction of a three-axis Computer Numeric Control(CNC) machine, for the purpose of rendering two dimensional vector graphics by using Inkscape .
- The CNC machine and control software will take vector image input in the form of Scalable Vector Graphics (SVG) files, and render the image onto a medium.
- The medium will be a flat surface, such as conventional paper, white board, or light reactive surface. The machine will be able to move on three axes and will be capable of drawing with multiple instruments, i.e. pencil, laser, etc. in an addition the machine will also be able to cut and plot on flat surface.
- The machine should meet the goals of balancing high precision and speed, use-limited resources and as many recycled parts as possible.
- Stepper motors or servo motors are used for precise motion. The working requirement of a router is larger which means requirement of longer ball screws. A good alternative can be to use belt for X and Y axes as they are cheap and requires less maintenance.
- A microcontroller reads the program file, interprets it and later sends it to the machine. ArduinoUNO is best suited to be used as a microcontroller since it is open source, cheap and readily available in the market. A Shield is inserted to the ArduinoUNO to connect the stepper drivers, limit switches and Emergency stop. Limit switches restrict the machine movement beyond the safe limit and an emergency stop is used to stop the machine in case of emergency

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose

Homework writing machine is an auto writing machine through which you can make your work easy by programing your project. As per the title this is a simple project using Arduino to make Homework writing machine at your home. This machine can draw any design and write any type of fonts. And this machine carving design on the wooden and acrylic sheet with high speed dc motor.

1.3.2 Scope

- This automated writing and drawing device is used to save the wastage of time.
- This machine will be able to carving and write the assignments and other hand work can save our time
- This machine can be used very easily for writing and carving we just need to give the input we need to give the measurement as the input and output.

1.3.3 Applicability

This project will help to many students are complete the assignment with in time of submission and the many have to start the small start-up business will be easy create carving On wooden, acrylic, metal sheet to and selling as per customer requirement both the work will be done at one machine

1.3.4 Achievement

The goal of the project is to construct a system that includes the principle of Mechanical and computer. The system should be user friendly for the operator. The main goals has been allocated

- Movement of X, Y, Z axis is controlled by motor which supplies either Alternating current or Direct current.
- Movement of machine is done by giving commands
- All the operations are carried out by codes like speed, feed, depth of cut, etc.

- For each operation separate code is available.

1.4 Organisation of Report

Chapter 1 contains the background and objective in a whole.

Chapter 2, the research of the project gets information from the Arduino website and github And the visiting the factory they work on cnc machine.

Chapter 3 system design

Chapter 4 implementation and testing code

Chapter 5 results and discussion

Chapter 6 conclusions and future work

Chapter 7 Bibliography

CHAPTER 2

SYSTEM ANALYSIS

2.1 Existing System

CNC machining is a manufacturing process in which pre-programmed computer software dictates the movement of factory tools and machinery. The process can be used to control a range of complex machinery, from grinders and lathes to mills and drill.

2.2 Purpose System

The basic idea of the homework writing machine is come from the CNC machine concept

All the working principal is the same (x, y, z) axis moment, the machine is smaller is size and easy to carry and software also user-friendly, the design is working is on both carving process and writing process it might be easy to remove pen holder and just add pcb drill just like plug and play no need any other specific work to do.

2.3 Requirements Specification

Prototype Model

- Requirements Gathering
- Quick design
- Building prototype
- Customer Evaluation
- Refining prototype
- Engineered prototype

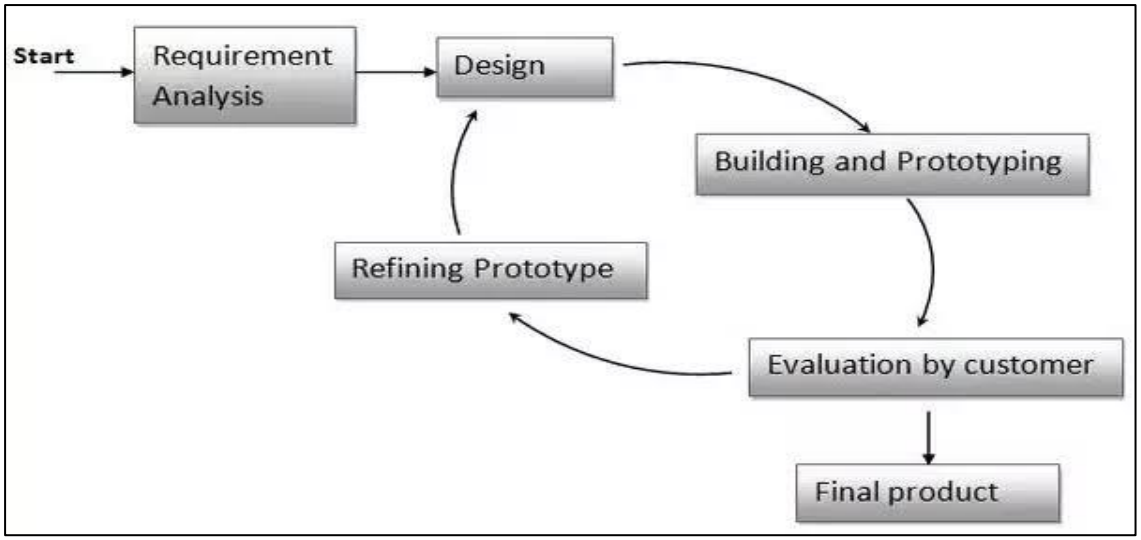

Fig.1 prototype model.

- **2.3.1 Requirements Gathering**

Knowing how to gather requirements is a skill that every analyst, and project manager, should have. However, it seems to be a skill that is generally lacking in many organizations. Poor requirements gathering is a major cause of project problems in many organizations The require information are collected for the github, official web site of Arduino and the visit the factory to analyzing the working principal Cnc machine The next step is where the "analyzing" starts. The purpose of validation is to make certain that the information conveyed during elicitation accurately represents the needs and expectations of the clients, the analyst prioritizes and formally documents the requirements in a Requirements Definition Report we prefer the basic cnc machine reports and video from the Google

- **2.3.2 Quick Design**

The quick design is the reference from the basic design of CNC machine with minor changes with the components and the smaller size from the original design. The design is the create with the help of AutoCAD software with dimension to rough idea of the machine looks like and the working and the adjustment the components in the design

- **2.3.3 Building Prototype**

Here we are assembling of the project with basic CAD design to get idea of the design from the reference & research And the concepts of machine is working given guide lines of the cnc machine to carving and designing

- **2.3.4 Customer Evaluation**

For checking the output of the given input by the customer to verify whether the work done is accurate or not. If the given output is not satisfied the prototype may have some problem. Then will make the changes on the prototype as per customer satisfaction, and the changes is starts from the design and coding create a new design and coding again this repeat process till the customer satisfied.

- **2.3.5 Refining Prototype**

After the costumer evaluation we are facing the problem with may be a design or coding section Here we know the Customer suggestion. We start working on the new prototype as per customer guideline the start the again the testing.

- **2.3.6 Final Product**

This is the final project complete is given to the customer require as per the given specification to the fulfill requirements and customer satisfaction simple and easy to use.

2.4 Hardware Requirements

1. Arduino
2. motor shield L293D
3. Steeper motor
4. Servo motor
5. Dc 12volt motor
6. Dc power supply (5,12) volt
7. Pen
8. Timing belt
9. Aluminium profile 30H and accessories
10. NUT and BOLT
11. Ball Bearing

2.5 Software Requirements

1. Arduino IDE
2. Inscape
3. SQL workbench
4. Processing 3.2.1 for G-code sender

2.6 Justification of Platform

Hardware Requirements Specification

2.6.1 Arduino Uno R3

The Arduino Uno R3 is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.
Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.
Revision 3 of the board has the following new features:

- 1.0 pin out: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

SUMMARY:-

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Table 1. Arduino Summary

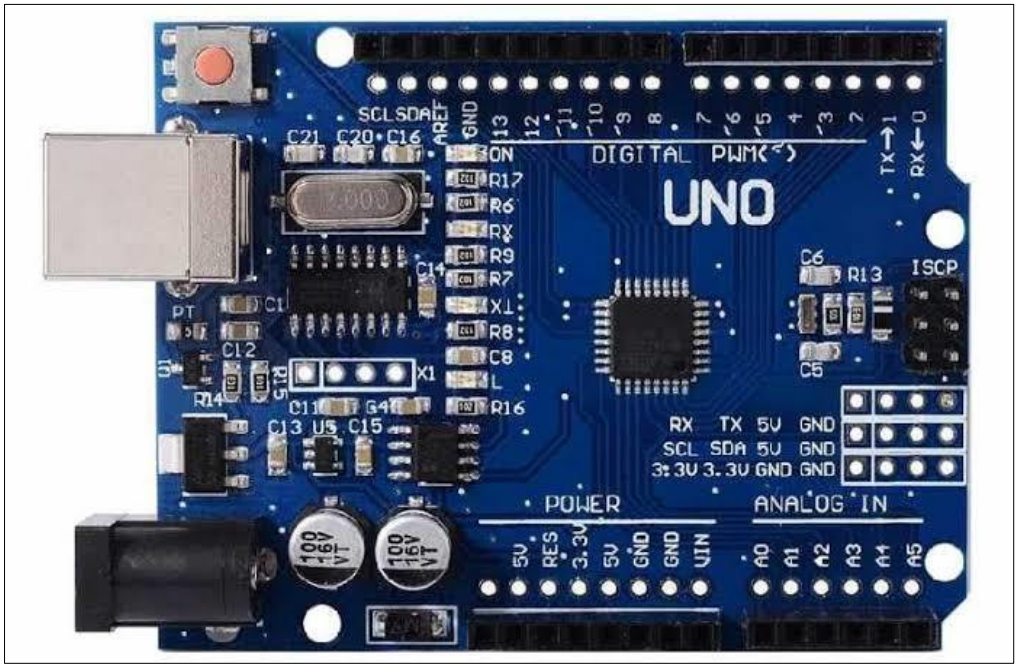

Fig.2 Arduino Uno R3

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digital Write(), and digital Read functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt () function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite () function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labelled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference () function. Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analog Reference ().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is

being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

2.6.2 L293d Motor Shield

Whenever a robotics hobbyist talk about making a robot, the first thing comes to his mind is making the robot move on the ground. And there are always two options in front of the designer whether to use a DC Motor or a stepper motor. When it comes to speed, weight, size, cost... DC motors are always preferred over stepper motors. There are many things which you can do with your DC motor when interfaced with a microcontroller. For example you can control the speed of motor, you can control the direction of rotation, you can also do encoding of the rotation made by DC motor i.e. keeping track of how many turns are made by your motors etc. So you can see DC motors are no less than a stepper motor. In this part of tutorial we will learn to interface a DC motor with a microcontroller. Usually H-bridge is preferred way of interfacing a DC motor. These days many IC manufacturers have H-bridge motor drivers available in the market like L293D is most used H-Bridge driver IC. H-bridge can also be made with the help of transistors and MOSFETs etc. rather of being cheap, they only increase the size of the design board, which is sometimes not required so using a small 16 pin IC is preferred for this purpose

FEATURES

- 8-bit serial input
- 8-bit serial or parallel output
- Storage register with 3-state output
- Shift register with direct clear
- 100 MHz (typical)shift out frequency
- ESD protection: HBM JESD22-A114F exceeds 2000V MM JESD22-A115-A exceeds 200v
- Multiple package option

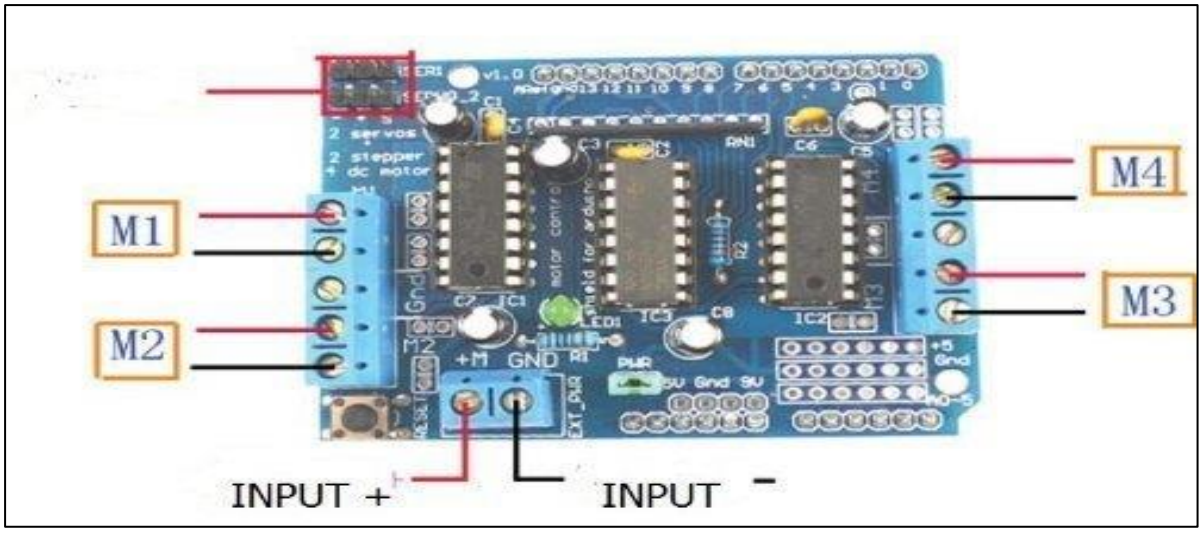

Fig.3 Motor Shield

- **L293d Dual H-Bridge Motor Driver**

L293 series of chips are power H-bridge motor drivers. The L293D chip is in 16-pins dip packages, and has two h-bridge drivers. An H bridge is typically capable of running two DC motor bidirectional (forward, backward), or two separate motors bidirectional. Thus a L293 chip can run two motors bidirectional, or 4 unidirectional.

Fig.4 L293D Dual H-Bridge Motor Driver

2.6.3 Stepper Motor

The stepper motors move in precisely repeatable steps, hence they are the motors of choice for the machines requiring precise position control. The Nema17 4.2 kg-cm Stepper motor can provide 4.2 kg-cm of torque at 1.7A current per phase. The motor's position can be commanded to move or hold at one position with the help of Stepper Motor Drivers. The Nema17 4.2 kg-cm Stepper motor provides excellent response to starting, stopping and reversing pulses from driver. They are very useful in the various applications, especially which demands low speed with high precision. Many machines such as 3D Printers, CNC Router and Mills, Camera Platforms, XYZ Plotters etc. It is a brushless DC motor, so the life of this motor is dependent upon life of the bearings. The position control is achieved by a simple Open Loop control mechanism so doesn't require complex electronic control circuitry. The motor's shaft has been machined for good grip with a pulley, drive gear etc. and especially avoiding stall or slip

Fig 5 stepper motors

Note**:**

1. The Nema17 4.2 kg-cm Stepper motor dimensions and weight may have ±2% error**.**
2. Resonances can occur because of improper installment.
3. Not easy to operate at extremely high speeds.

Features*:*

1. Input pulse decides the rotation angle of the motor.
2. High accuracy of around 3 to 5% a step.
3. Provides good starting, stopping and reversing.
4. Control of this motor is less costly because of the exclusion of complex control circuitry.
5. The speed is proportional to the frequency of the input pulses.

2.6.4 Servo Motor

The Tower Pro MG995 High-Speed Digital Servo Motor rotates 90° in each direction making it 180° servo motor. It is a Digital Servo Motor which receives and processes PWM signal faster and better. It equips sophisticated internal circuitry that provides good torque, holding power, and faster updates in response to external forces.

They are packed within a tight sturdy plastic case which makes them water and dust resistant which is a very useful feature in RC planes, Boats, and RC Monster Trucks etc. It equips 3-wire JR servo plug which is compatible with Futaba connectors too.

Fig.6 Servo Motor

Wire Description

- **RED –** Positive
- **Brown –** Negative
- **Orange –** Signal

Features:

- **The connection cable is thicker.**
- **Equips high-quality motor.**

- High resolution
- Accurate positioning
- Fast control response
- Constant torque throughout the servo travel range
- Excellent holding power

2.6.5 Dc 12volt Motor

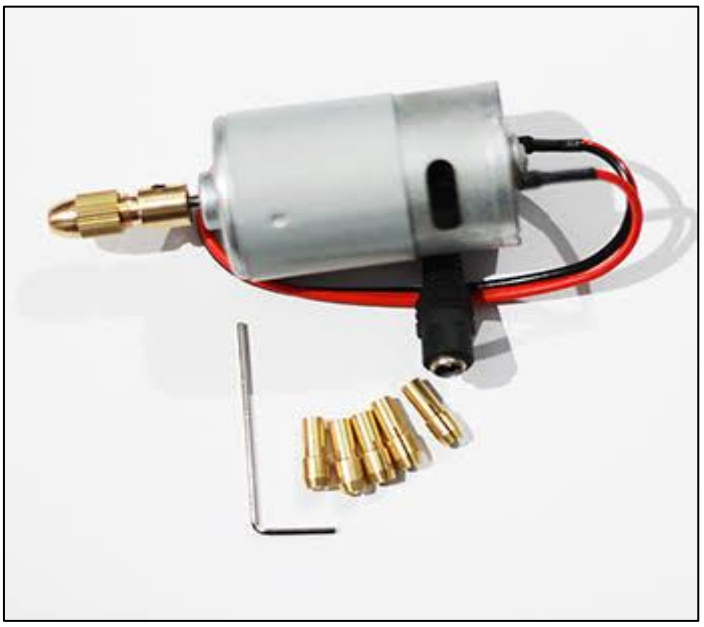

Fig.7 DC motor

A DC motor is any motor within a class of electrical machines whereby direct current electrical power is converted into mechanical power. Most often, this type of motor relies on forces that magnetic fields produce. Regardless of the type, DC motors have some kind of internal mechanism, which is electronic or electromechanical. In both cases, the direction of current flow in part of the motor is changed periodically mostly motor is use toys and PCB drill machine

2.6.6 DC power supply 5 volt & 12 volt

Fig 8 power supply

5V power supplies (or 5VDC power supplies) are one of the most common power supplies in use today. Linear regulated 5VDC power supplies regulate the output using a dissipative regulating circuit. Its use for external power supply to the motor shield to run the stepper motor 12 volt power supply is use for the dc motor for engraving purpose the dc motor is maximum power on 12volt the motor has higher rpm and work fine

2.6.7 Pen

Fig.9 pen

A pen is a writing instrument used to apply ink to a surface, usually paper, for writing or drawing. Historically, reed pens, quill pens, and dip pens were used, with a nib dipped in ink. … Modern types include ballpoint, rollerball, fountain and felt or ceramic tip pens

2.6.8 Timing Belt

Fig.10 timing belt

The Timing Belt is also called as the synchronous belt is popularly known for its non-slipping mechanical drive belt. It is composed of the flexible belt which contains a row of teeth embedded on the inner surface of the belt. Timing Pulley and belt works when the toothed parts become compatible with each other**.** It is a belt that usually features teeth on the inside surface, while a timing chain is a roller chain. Widely used in mechanical devices, printers/photocopiers, 3d printer, robotics, automation, etc.

Features

1. Precision registration and timing with no loss of high torque carrying capability
2. Minimal vibration and chordal effect Positive slip proof engagement
3. Wide speed range, especially important when the entire speed range is developed from a single source
4. Virtually no elongation (stretching) due to wear
5. Power transmission efficiency is not lost with the use
6. A clean operation, no need for lubrication
7. Reduced noise.
8. Long, dependable trouble-free service.

2.6.9 Aluminium Profile 30h and Accessories

Fig.11 ALUMINIUM PROFILE 30H

From the robust aluminum square profile, the versatile Eco Shape round tubes, through to linear guides and a wide range of connection solutions: The modular profile system currently includes more than 1,500 modular components for all areas of your production – even for ESD, clean and dry room applications. All components are finely coordinated and offer you an almost infinite number of combination possibilities when designing your working environments. The aluminum profiles are natural anodized and therefore permanently scratch resistant and corrosion proof. The circumferential anodized coating makes the saw cut particularly low burr.

- **Aluminium Profile T-Nut**

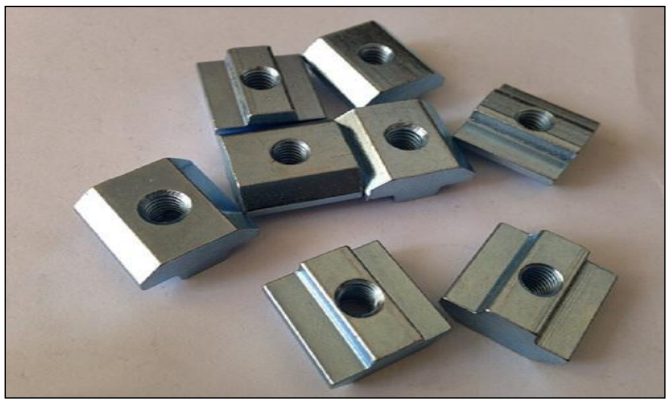

FIG.12 T-NUT

Light and strong, our Aluminum extrusion product - T-Profile T nut For Aluminum profile, Size: Nm 6, Nm 8 has excellent resistance to corrosion. This is a perfect choice for wide range of construction or manufacturing purpose. .The product is ideal for use in shelving, craft modeling projects.

- **Aluminum Profile L-Joint**

Fig. 13 ALUMINUM PROFILE L-JOINT

The L shape hidden corner bracket used for setting up a right angle connection of the two aluminium profiles, insert the two ends of the L bracket into the groove of the profile, and tighten the set screw to fix the two profiles. Concealed corner brace, the L corner bracket can be completely hidden in the slot of the aluminium extrusions Widely used for building industrial aluminium frame structures, 3D printers, CNC Routers, CNC laser cutters, Robotics Projects.

2.7.0.0 Nut and Bolt

Fig 14 NUTAND BOLT

A screw thread is a helical groove on a shaft. When used for delivering power, it is called a drive screw. Drive screws aren't really all that efficient, as they loose a significant amount of power to friction. However, this friction can be put to use in the case of threaded fasteners. You might say that a drive screw is an inclined plane wrapped around a post, while a fastener is a wedge wrapped around a post.

2.7.0.1 Ball Bearing

Fig 15 Ball Bering

A bearing in which the parts are separated by a ring of small freely rotating metal balls which reduce friction The steel balls act as ball bearings so that the action is smoother than just one gear turning against another.

2.7.0.2 Arduino IDE

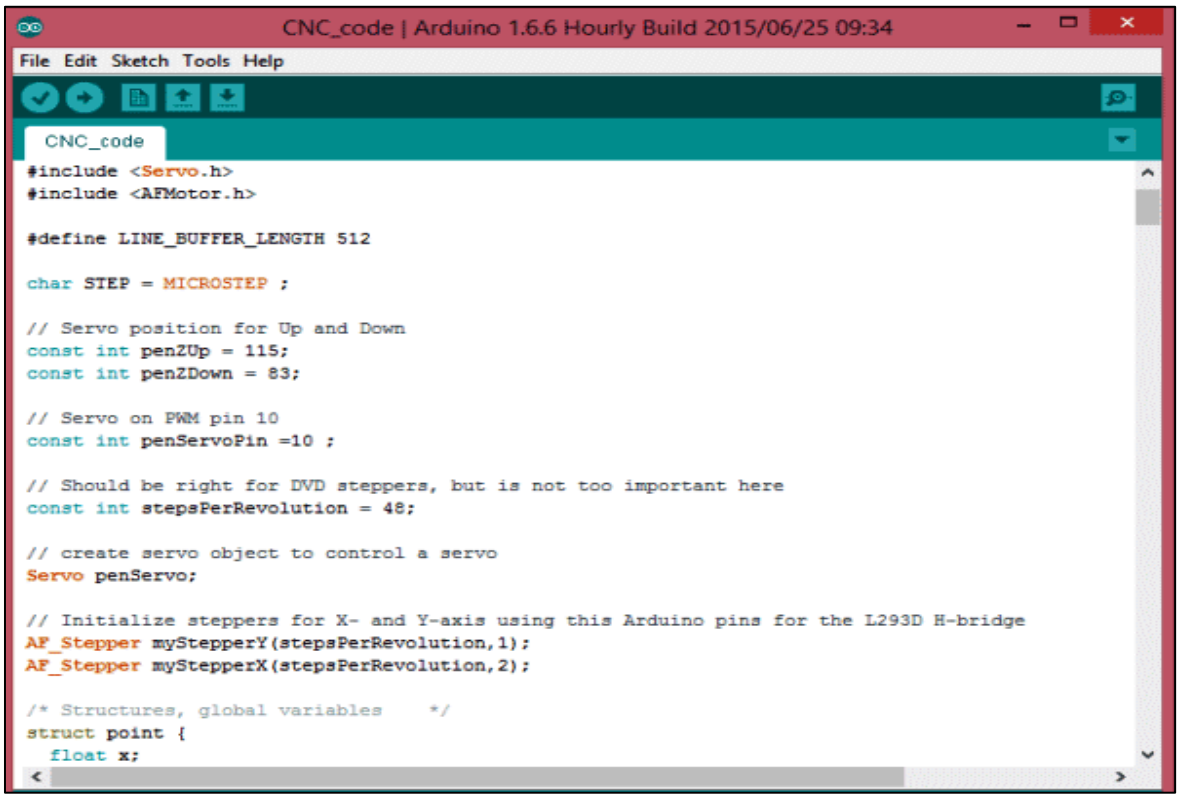

Fig.16 Arduino IDE

Arduino integrate development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards

The source code for the IDE is released under the GNU General Public License, version 2.The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware

2.7.0.3 Inkscape

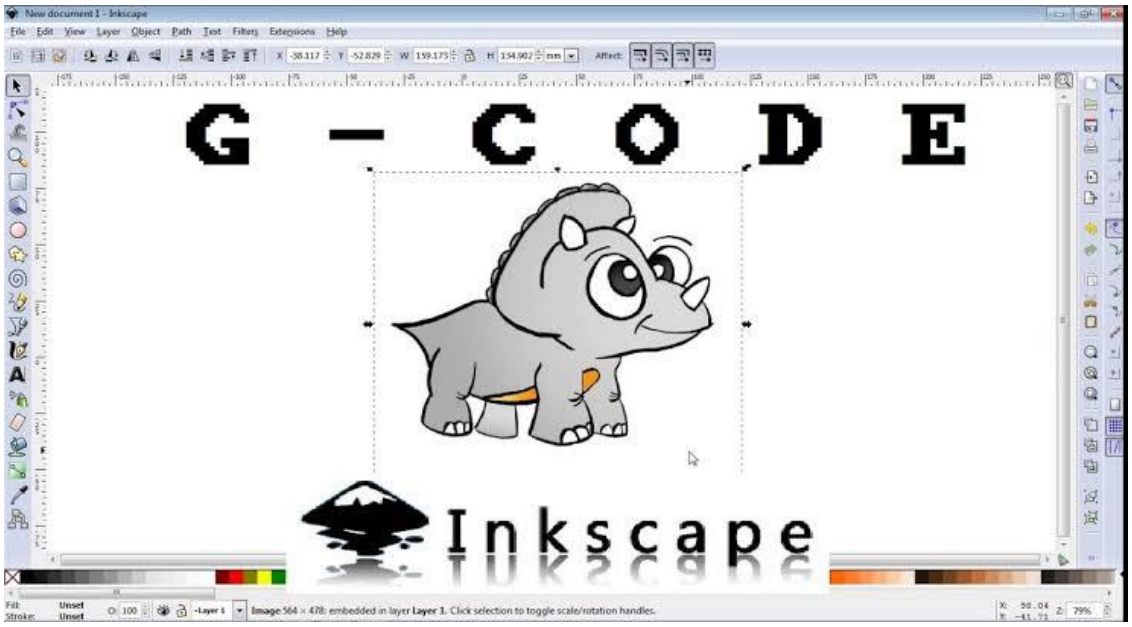

Fig.17 Inkscape

Inkscape is a free and open-source vector graphics editor. This software can be used to create or edit vector graphics such as illustrations, diagrams, line arts, charts, logos and complex paintings. Inkscape's primary vector graphics format is Scalable Vector Graphics (SVG); however, many other formats can be imported and exported. Inkscape can render primitive vector shapes (e.g. rectangles, ellipses, polygons, arcs, spirals, stars and 3D boxes) and text. These objects may be filled with solid colours, patterns, radial or linear colour gradients and their borders may be stroked, both with adjustable transparency. Embedding and optional tracing of raster graphics is also supported, enabling the editor to create vector graphics from photos and other raster sources. Created shapes can be further manipulated with transformations, such as moving, rotating, scaling and skewing. We have to add libraries files to image into g-code So we choose the document properties of the Inkscape 40cmx40cm (Width × Height) which is four times the working area of the plotter because the plotter can draw only in the first quadrant. So we have initially kept the axes at the nearest end of the motors which is considered as origin to easily modify the design. In Fig. 3 the working area of CNC plotter is shown with the text written in the pre-defined area. The text is selected using cursor and then select "object to path" from the drop down window to save the G-code form of the selected text. To create G-code of an image, the file must have a transparent background. The image should be dragged into the selected area then select "trace bitmap" from drop down window to create a transparent image. Scans are selected as 8 and

“Edge detection” is selected to create black & white image. After adding this transparent image in the predefined area we’ve used “object to path” command to create the G-code file of the selected image by following the steps described earlier

2.7.0.4 MySQL Workbench

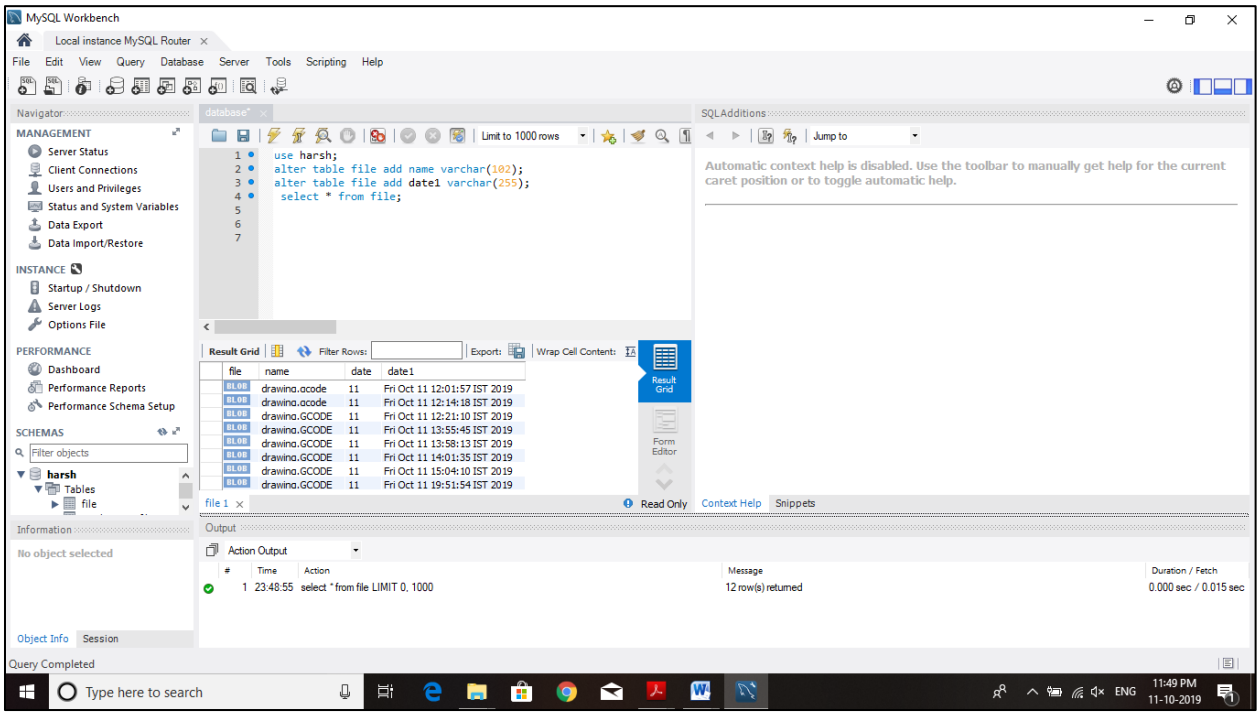

Fig.18 MySQL Workbench

MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system

2.7.0.5 Processing 3.2.1 for G-Code Sender

Processing is an open-source graphical library and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching non-programmers the fundamentals of computer programming in a visual context.

Processing uses the Java language, with additional simplifications such as additional classes and aliased mathematical functions and operations. It also provides a graphical user interface for simplifying the compilation and execution stage. The Processing

language and IDE were the precursor to other projects including Arduino, processing IDE upload the g code in to Arduino cnc machine

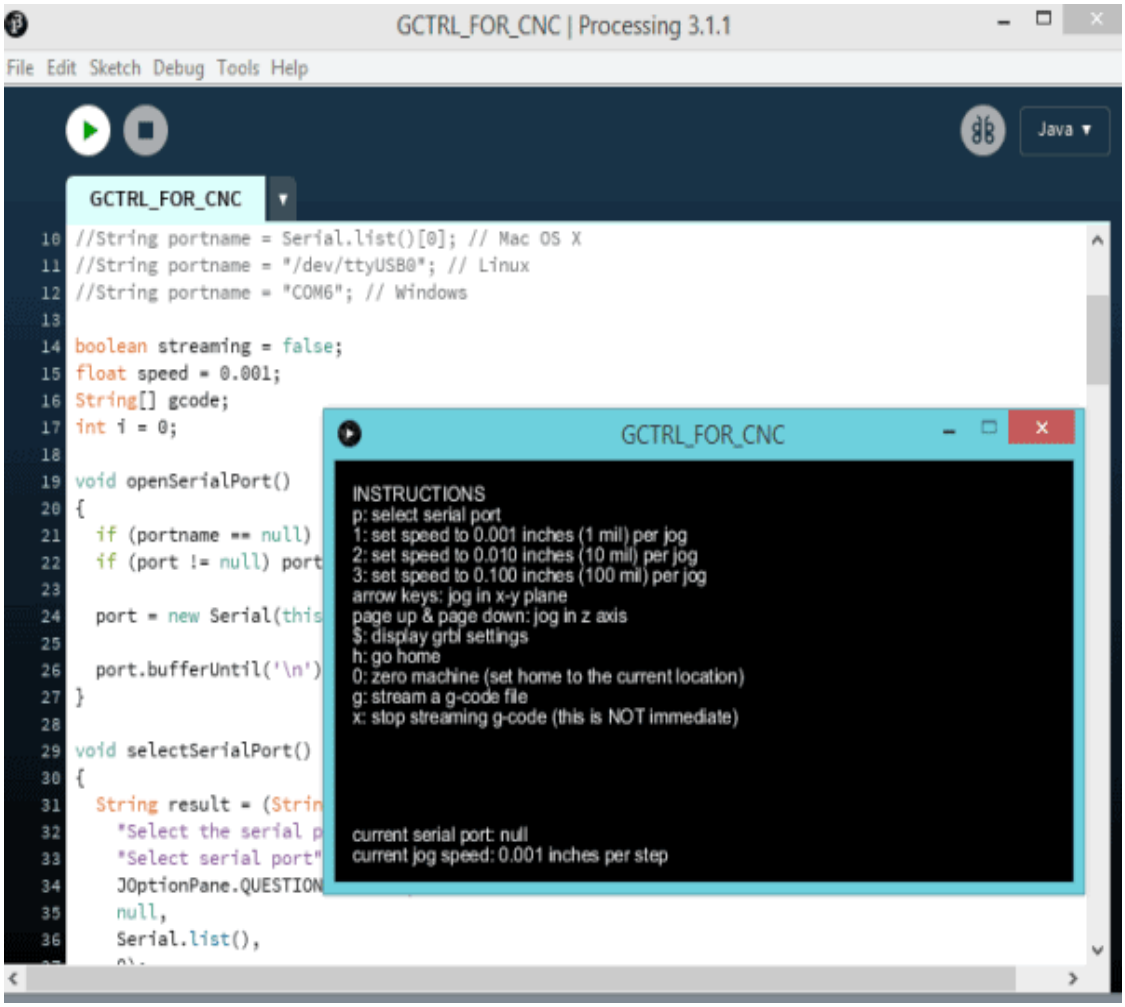

Fig.19 processing IDE

CHAPTER 3

SYSTEM DESIGN

3.1 Module Division

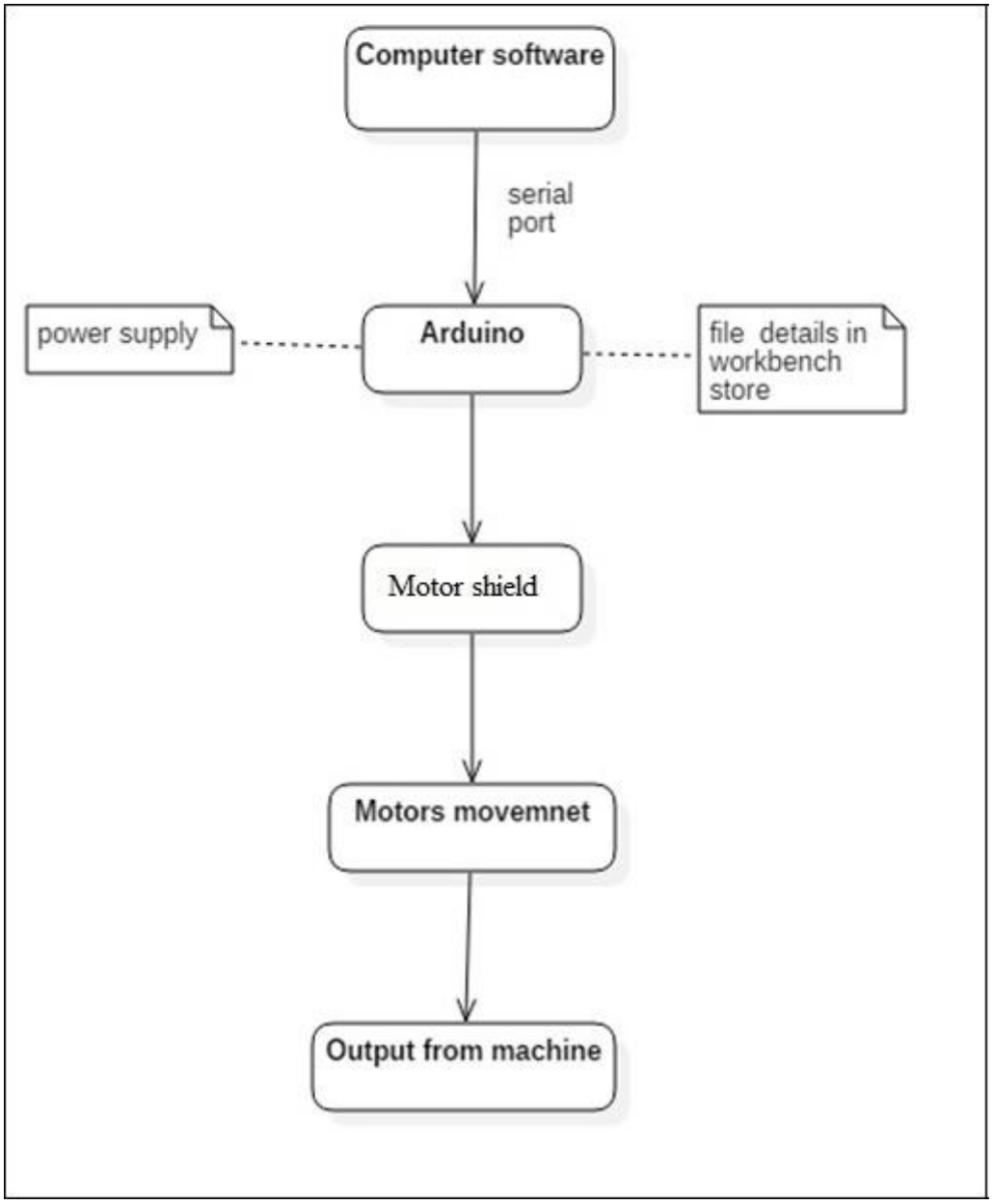

Fig.20 MODULES DIVISION

- **Computer Software**

The software converts the image to g code and first programme to Arduino for cnc work

And the send to data from the serial port and the file details will be save on the MYSQL

- **Arduino**

It’s a microcontroller to work as per specific task and the given instruction

Power supply is mandatory and the input file details store in sql workbench

- **Motor Shield**

The motor shield is the control the motor forward and backward

- **Motor Movement**

The motor movement as the given g-code coordinates

- **Output From Machine**

As per requirement form the user we use pen or PCB drill

3.2 Data Dictionary

- **Schema name :- Cnc**

FILE NAME	DATA TYPE	MEASUREMENT	CONSTRAINTS
Id	**Integer**	**Auto increment**	**Not null primary key**
Name	**Varchar**	**102**	**allow null**
Date	**Varchar**	**255**	**Not null**

Table .2 information of the file

- **Data Integrity**

  First the check the g-code file is create as per given design from the user, upload g-code in the Arduino that time automatically file name and timing of upload details save in the database we can check the details in show in MySQL workbench

3.3 Entity Relationship Diagram

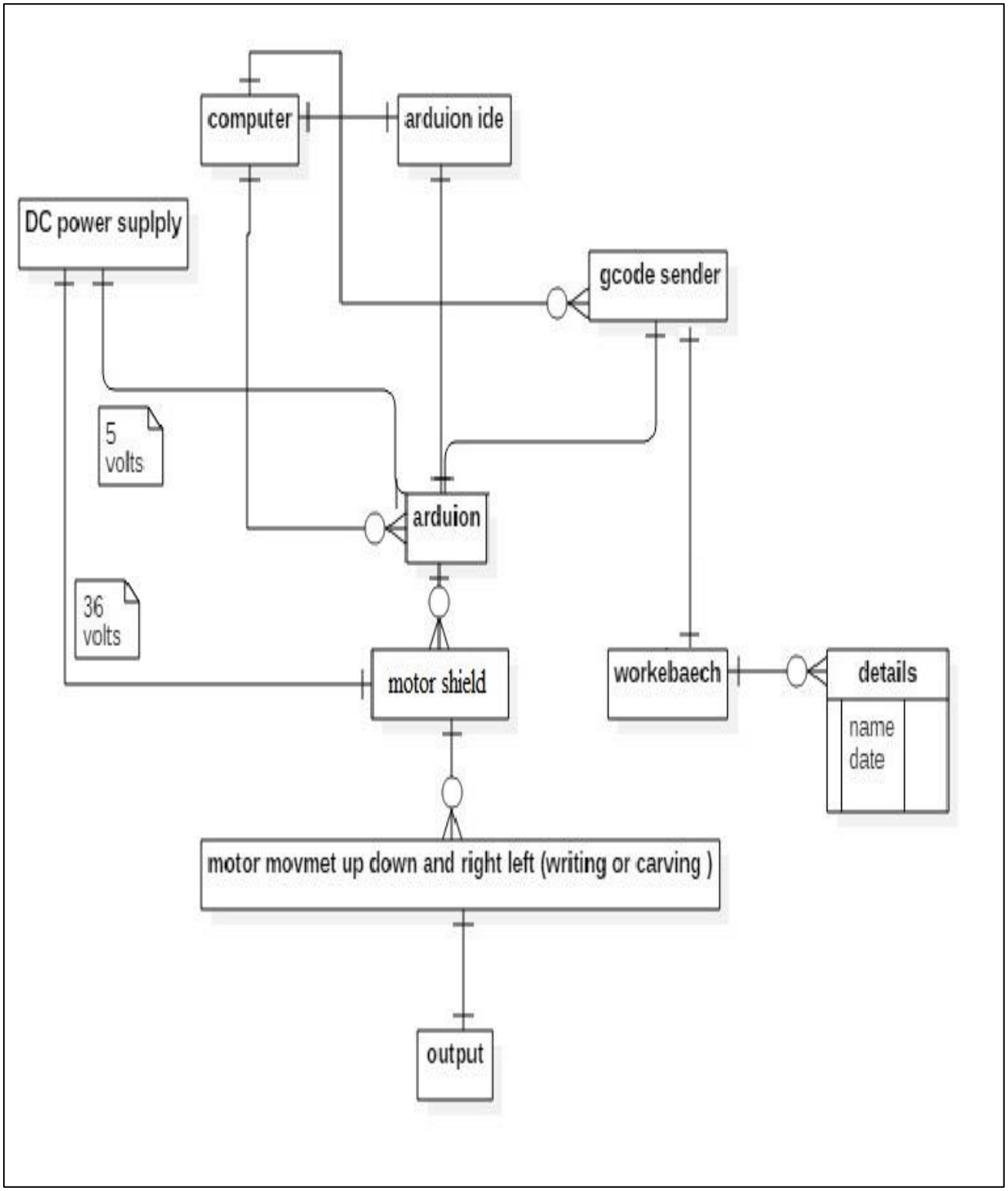

Fig.21 Entity Relationship Diagram

The given diagram is line the relation between the entity and the next entity one to one relation and one to one relationship between and they also give a flow diagram of working the machine

3.4 Data Flow Diagram

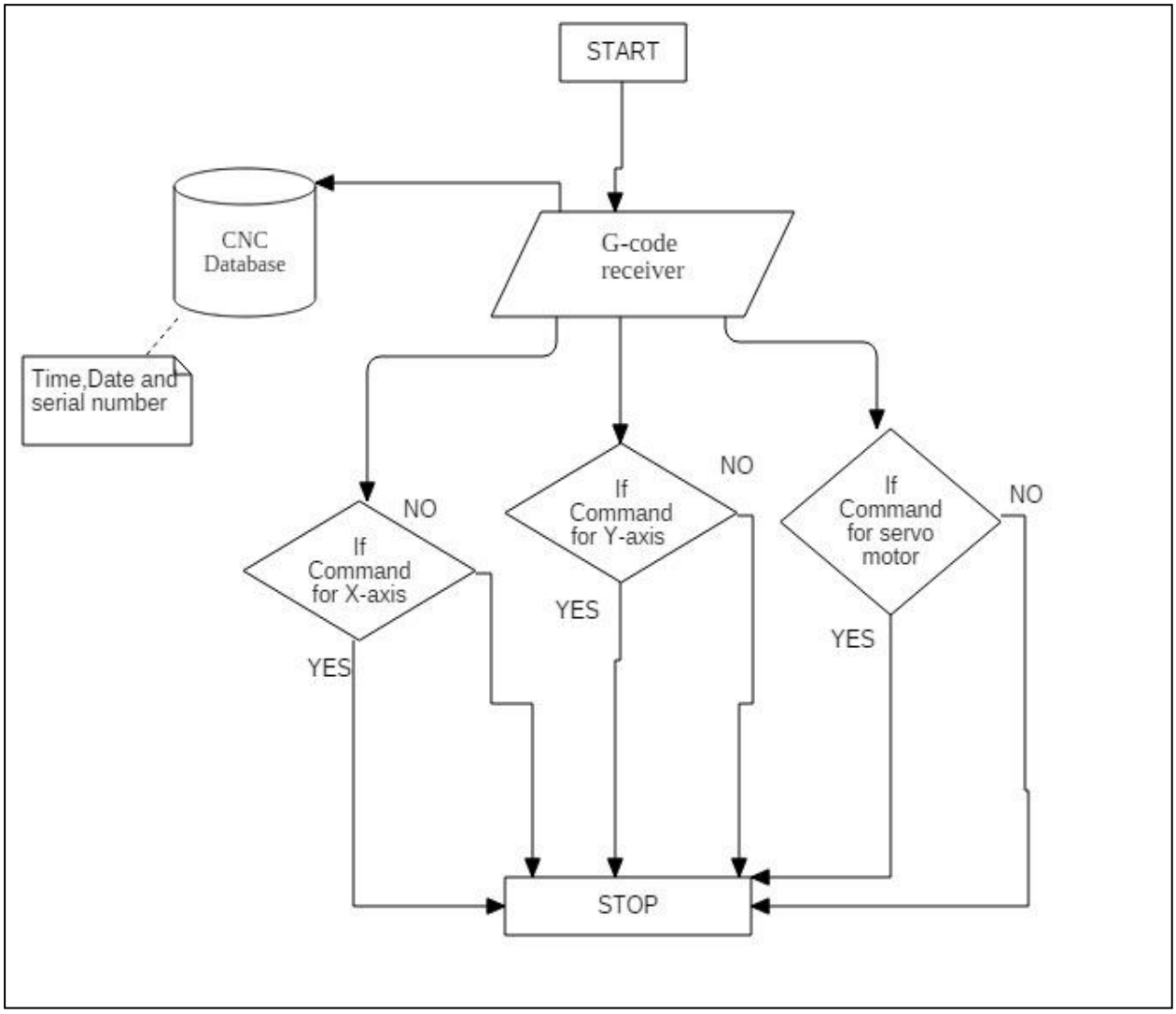

Fig.22 Data Flow Diagram

This diagram is show the way the data will flow in the machine work if any coordinates are missing in the g code the machine will stop working

3.5 Class Diagram

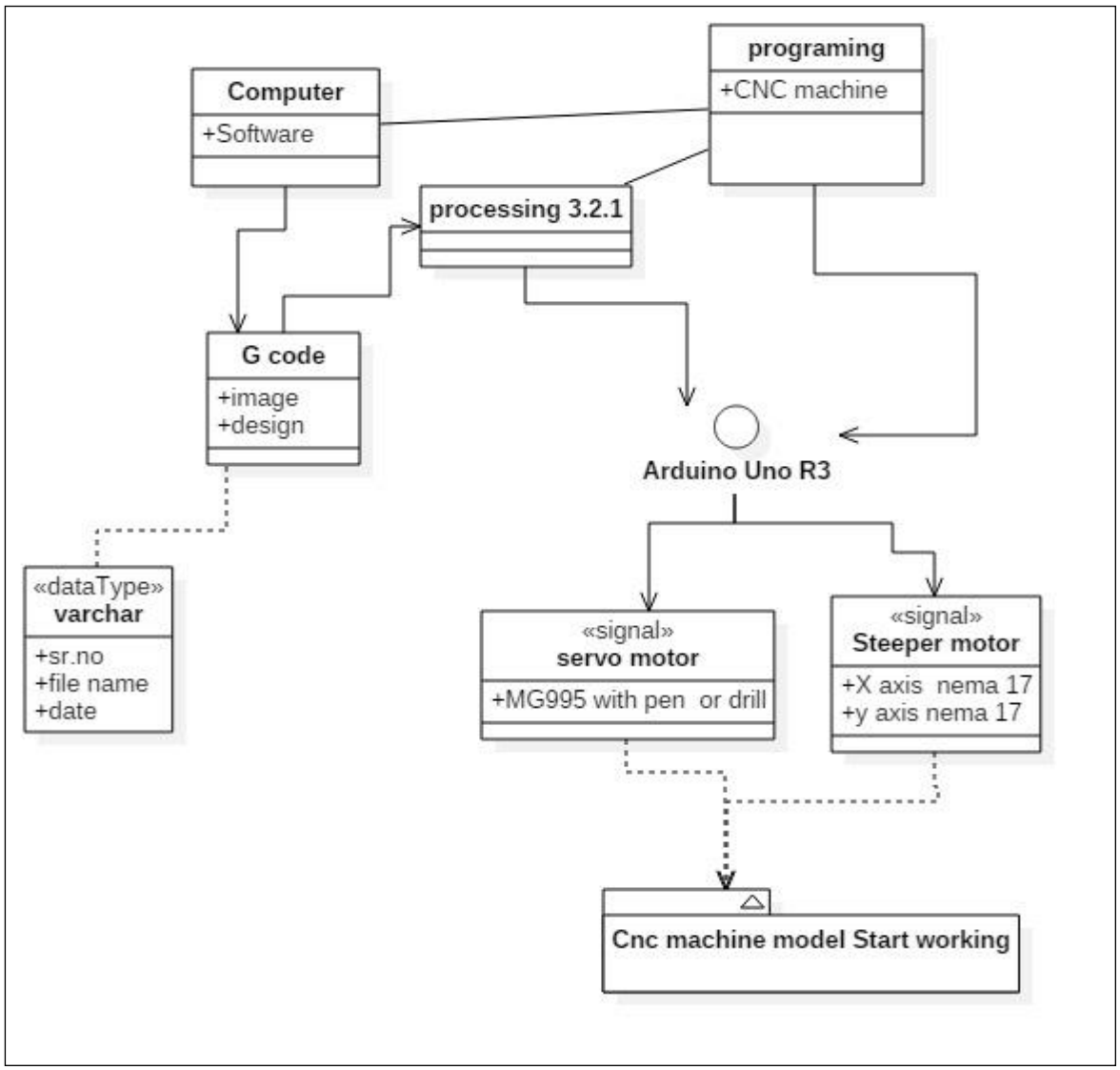

Fig 23 Class diagram

This diagram is show that the all the attribute are use in this project

3.6 Component Diagram

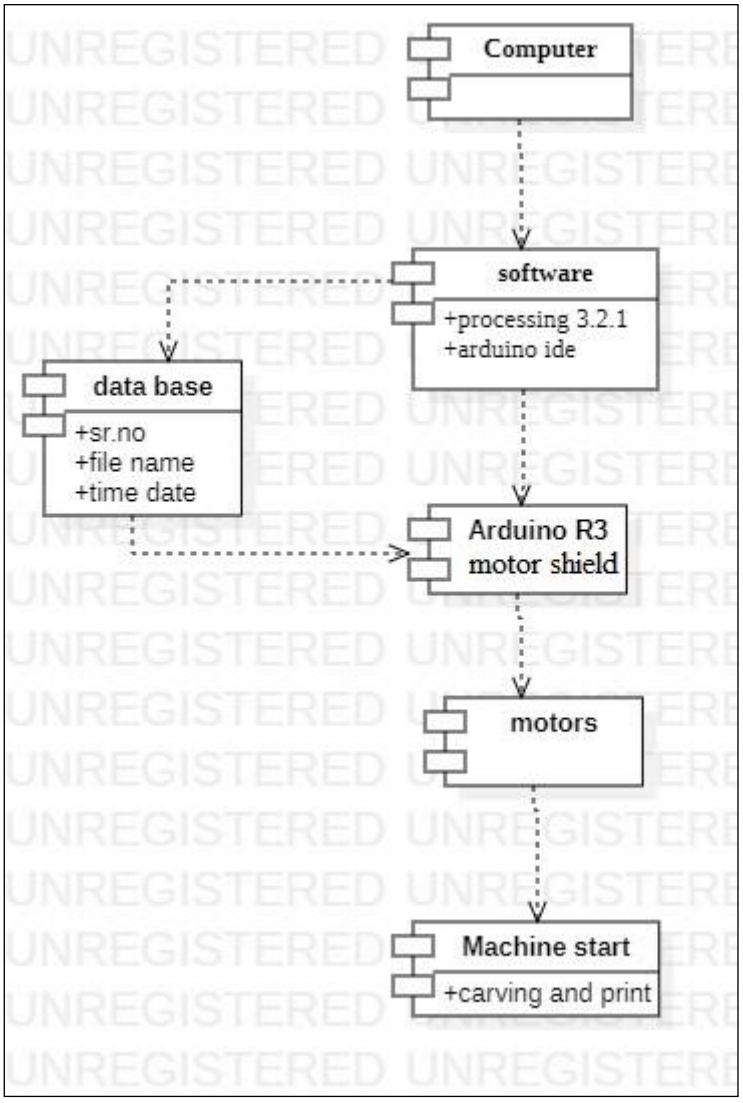

Fig 24 Component diagram

3.7 Deployment Diagram

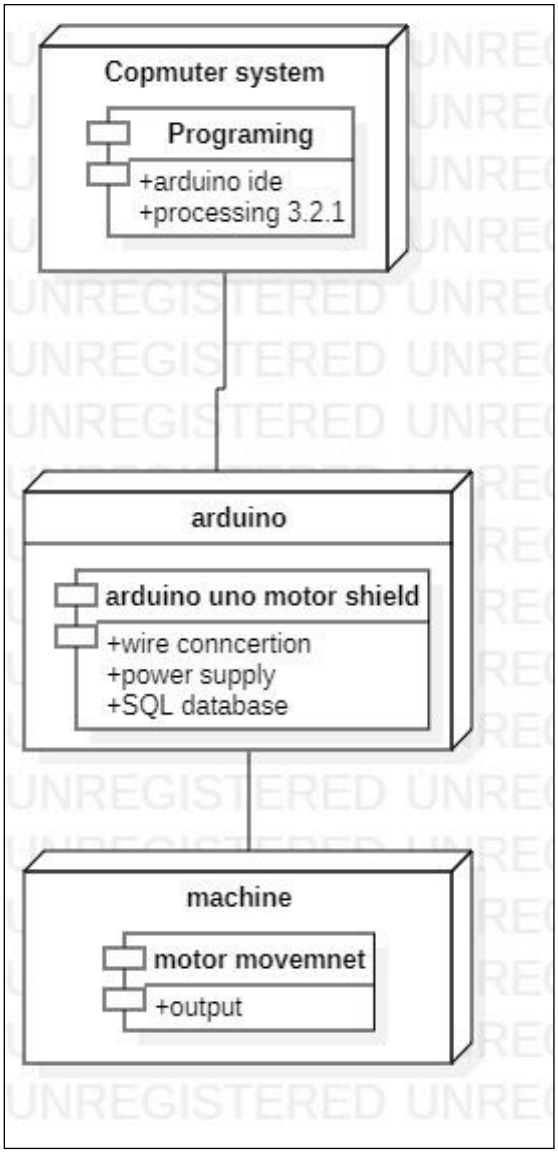

Fig 25 Deployment diagram

3.8 User Interface Design

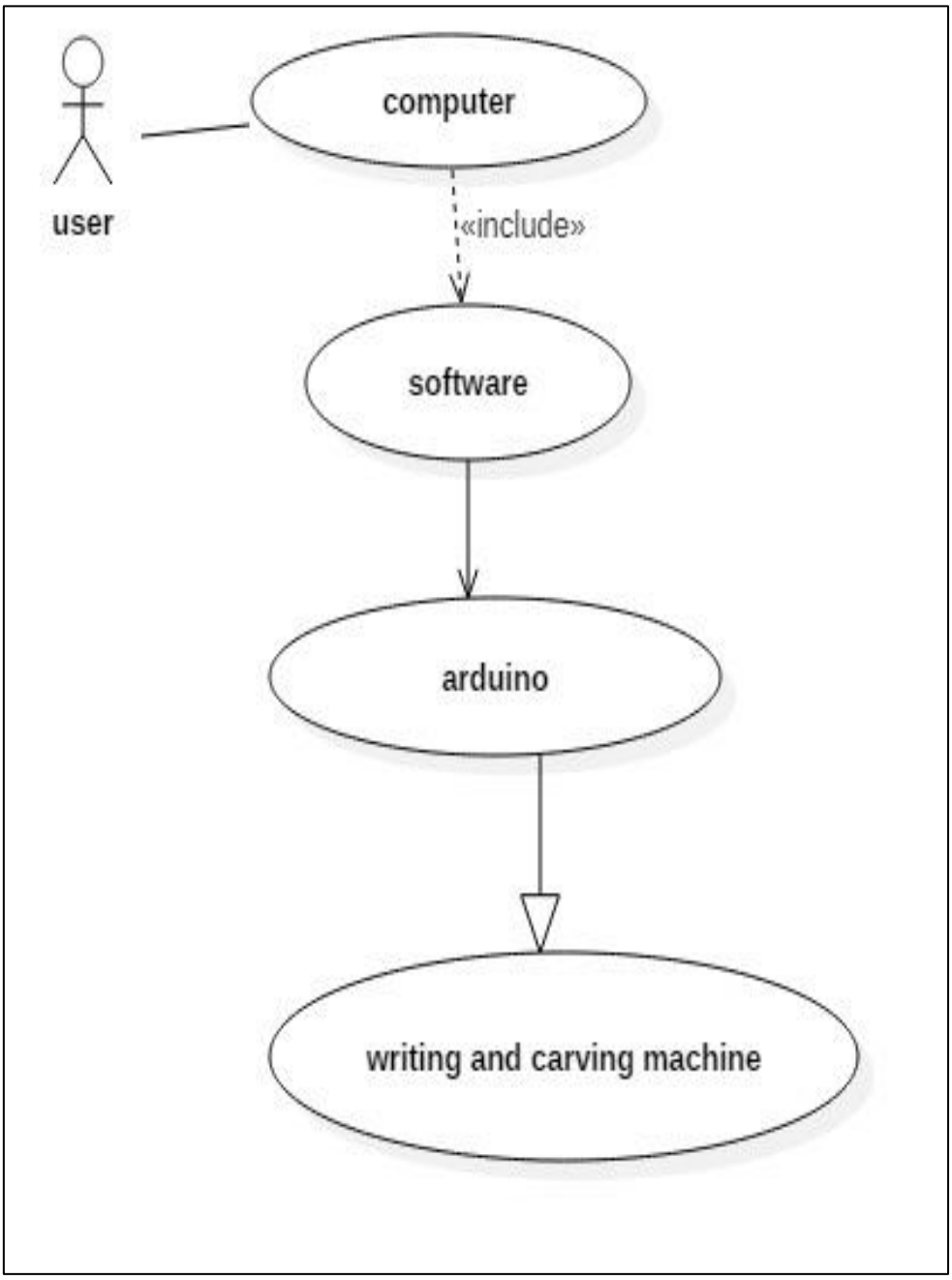

Fig.26 User Interface Design

The user is given any input require the computer or to first convert the image into g code and after send through the computer then the machine start working as per given code

3.9 Procedural Design

The design the machine step by step given details

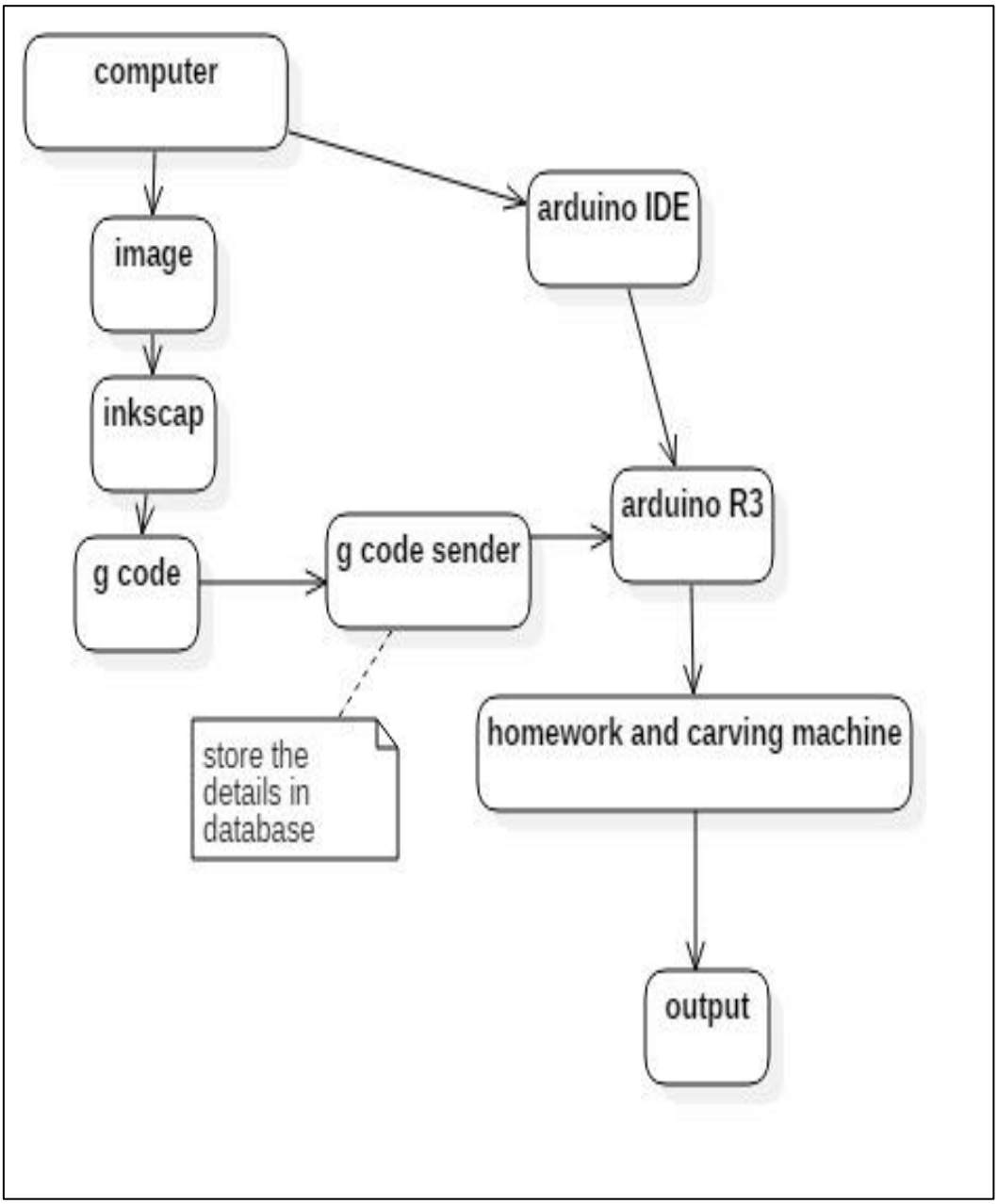

Fig 27 Procedural Design

Image converts them into g code form the inscape software and save the file into g code and the Arduino programme for cnc working next step send the g code to the machine that time g code file details will save on the database and machine will start working.

3.4.0 Wiring Design

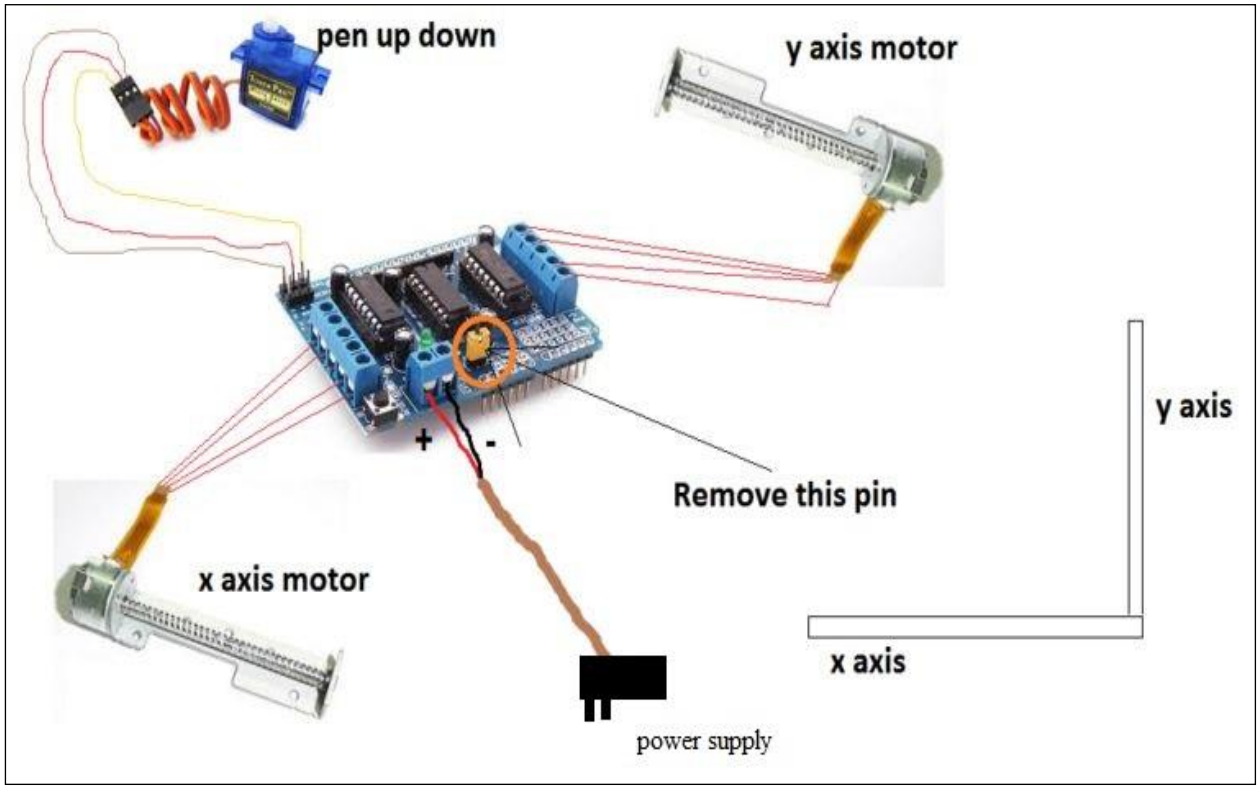

Fig.28 Wiring Design

The given information into image all the components connectivi1ty properly and check the motor connections same colour code wiring all the side jumpers to connect the enable pots and clock signals

3.4.1 Sequence Diagram

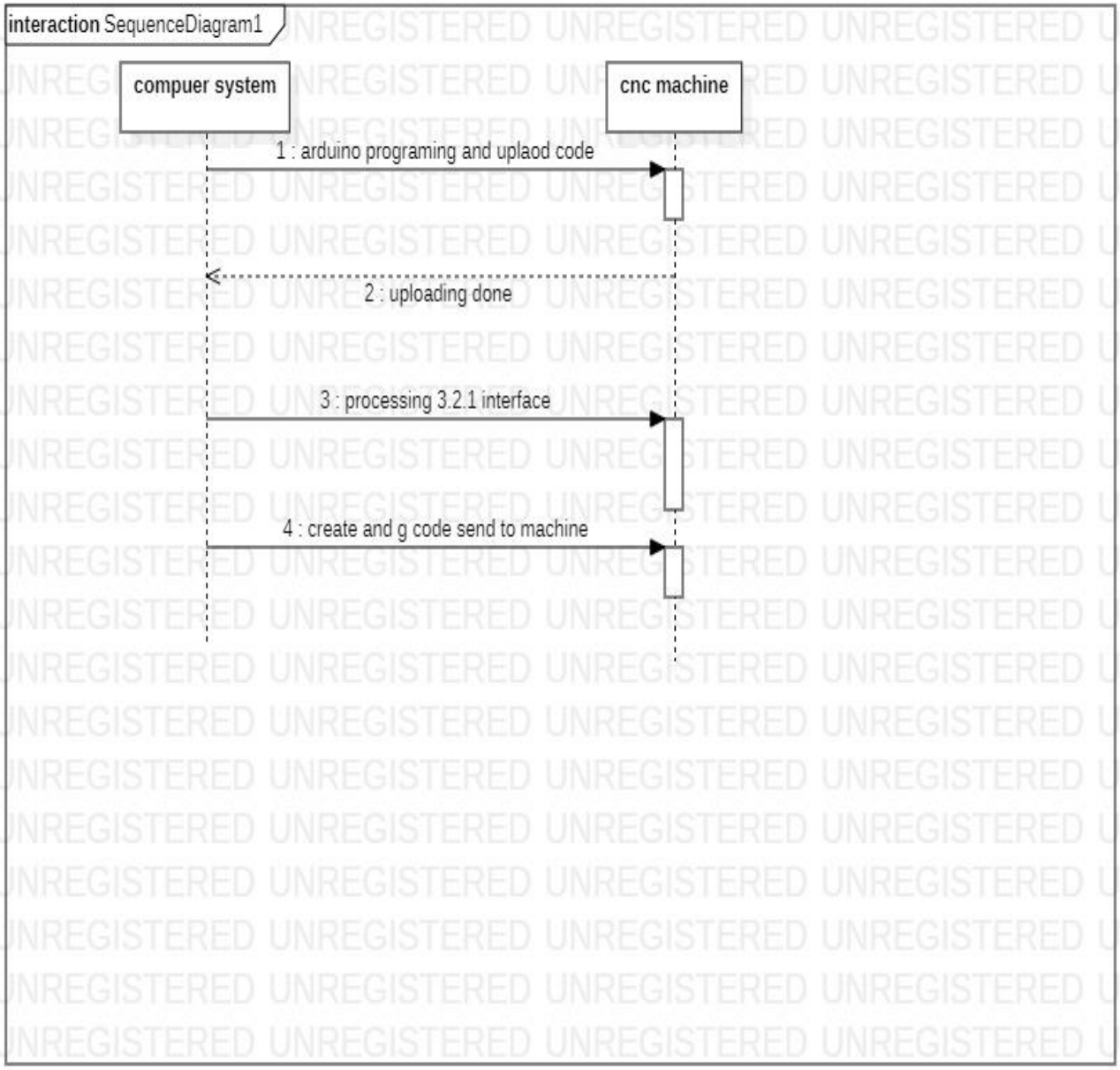

Fig.29 Sequence Diagram

This diagram of the working step by step working

CHAPTER 4

IMPLEMENTATION AND TESTING

4.1 Coding

4.1.1 CNC machine code in C++

```
/*add the library function */
#include <Servo.h>
#include <AFMotor.h>
#define LINE_BUFFER_LENGTH 1024
char STEP = MICROSTEP ;
// Servo position for Up and Down
const int penZUp = 115;
const int penZDown = 83;
// Servo on PWM pin 10
const int penServoPin =10 ;
// Should be right for nema 17 steppers motor, but is not too important here
const int stepsPerRevolution = 4096;
// create servo object to control a servo
Servo penServo;
// Initialize steppers for X- and Y-axis using this Arduino pins for the L293D H-bridge
AF_Stepper myStepperY(stepsPerRevolution,1);
AF_Stepper myStepperX(stepsPerRevolution,2);
/* Structures, global variables    */
struct point {
float x;
float y;
float z;
};
```

```
// Current position of plothead
struct point actuatorPos;
//  Drawing settings, should be OK
float StepInc = 1;
int StepDelay = 0;
int LineDelay =0;
int penDelay = 50;
// Motor steps to go 1 millimeter.
// Use test sketch to go 100 steps. Measure the length of line.
// Calculate steps per mm. Enter here.
float StepsPerMillimeterX = 100.0;
float StepsPerMillimeterY = 100.0;
// Drawing robot limits, in mm
// OK to start with. Could go up to 50 mm if calibrated well.
float Xmin = 0;
float Xmax = 300;
float Ymin = 0;
float Ymax = 300;
float Zmin = 0;
float Zmax = 1;
float Xpos = Xmin;
float Ypos = Ymin;
float Zpos = Zmax;
// Set to true to get debug output.
boolean verbose = false;
//  Needs to interpret
//  G1 for moving
//  G4 P300 (wait 150ms)
//  M300 S30 (pen down)
```

```
//  M300 S50 (pen up)
//  Discard anything with a (
//  Discard any other command!
/**********************
* void setup() - Initialisations
***********************/
void setup() {
//  Setup
Serial.begin( 9600 );
penServo.attach(penServoPin);
penServo.write(penZUp);
delay(100);
// Decrease if necessary
myStepperX.setSpeed(10000);
myStepperY.setSpeed(10000);
//  Set & move to initial default position
// TBD
//  Notifications!!!
Serial.println("Mini CNC Plotter alive and kicking!");
Serial.print("X range is from ");
Serial.print(Xmin);
Serial.print(" to ");
Serial.print(Xmax);
Serial.println(" mm.");
Serial.print("Y range is from ");
Serial.print(Ymin);
Serial.print(" to ");
Serial.print(Ymax);
Serial.println(" mm.");
```

```
}
/**********************
* void loop() - Main loop
**********************/
void loop()
{
delay(100);
char line[ LINE_BUFFER_LENGTH ];
char c;
int lineIndex;
bool lineIsComment, lineSemiColon;
lineIndex = 0;
lineSemiColon = false;
lineIsComment = false;
while (1) {
// Serial reception - Mostly from Grbl, added semicolon support
while ( Serial.available()>0 ) {
c = Serial.read();
if (( c == '\n') || (c == '\r') ) {             // End of line reached
if ( lineIndex > 0 ) {                        // Line is complete. Then execute!
line[ lineIndex ] = '\0';                   // Terminate string
if (verbose) {
Serial.print( "Received : ");
Serial.println( line );
}
processIncomingLine( line, lineIndex );
lineIndex = 0;
}
else {
```

```
// Empty or comment line. Skip block.
}
lineIsComment = false;
lineSemiColon = false;
Serial.println("ok");
}
else {
if ( (lineIsComment) || (lineSemiColon) ) {   // Throw away all comment characters
if ( c == ')' )  lineIsComment = false;     // End of comment. Resume line.
}
else {
if ( c <= ' ' ) {                   // Throw away whitepace and control characters
}
else if ( c == '/' ) {                  // Block delete not supported. Ignore character.
}
else if ( c == '(' ) {         // Enable comments flag and ignore all characters until ')' or EOL
lineIsComment = true;
}
else if ( c == ';' ) {
lineSemiColon = true;
}
else if ( lineIndex >= LINE_BUFFER_LENGTH-1 ) {
Serial.println( "ERROR - lineBuffer overflow" );
lineIsComment = false;
lineSemiColon = false;
}
else if ( c >= 'a' && c <= 'z' ) {        // Upcase lowercase
line[ lineIndex++ ] = c-'a'+'A';
}
```

```
else {
line[ lineIndex++ ] = c;
}
}
}
}
}
}
void processIncomingLine( char* line, int charNB ) {
int currentIndex = 0;
char buffer[ 128 ];                    // Hope that 64 is enough for 1 parameter
struct point newPos;
newPos.x = 0.0;
newPos.y = 0.0;
//  Needs to interpret
//  G1 for moving
//  G4 P300 (wait 150ms)
//  G1 X60 Y30
//  G1 X30 Y50
//  M300 S30 (pen down)
//  M300 S50 (pen up)
//  Discard anything with a (
//  Discard any other command!
while( currentIndex < charNB ) {
switch ( line[ currentIndex++ ] ) {          // Select command, if any
case 'U':
penUp();
break;
case 'D':
```

```
penDown();
break;
case 'G':
buffer[0] = line[ currentIndex++ ];          // /!\ Dirty - Only works with 2 digit commands
//      buffer[1] = line[ currentIndex++ ];
//      buffer[2] = '\0';
buffer[1] = '\0';

switch ( atoi( buffer ) ){                  // Select G command
case 0:                                     // G00 & G01 - Movement or fast movement. Same here
case 1:
// /!\ Dirty - Suppose that X is before Y
char* indexX = strchr( line+currentIndex, 'X' );  // Get X/Y position in the string (if any)
char* indexY = strchr( line+currentIndex, 'Y' );
if ( indexY <= 0 ) {
newPos.x = atof( indexX + 1);
newPos.y = actuatorPos.y;
}
else if ( indexX <= 0 ) {
newPos.y = atof( indexY + 1);
newPos.x = actuatorPos.x;
}
else {
newPos.y = atof( indexY + 1);
indexY = '\0';
newPos.x = atof( indexX + 1);
}
drawLine(newPos.x, newPos.y );
//        Serial.println("ok");
```

```
actuatorPos.x = newPos.x;
actuatorPos.y = newPos.y;
break;
}
break;
case 'M':
buffer[0] = line[ currentIndex++ ];        // /!\ Dirty - Only works with 3 digit commands
buffer[1] = line[ currentIndex++ ];
buffer[2] = line[ currentIndex++ ];
buffer[3] = '\0';
switch ( atoi( buffer ) ){
case 300:
{
char* indexS = strchr( line+currentIndex, 'S' );
float Spos = atof( indexS + 1);
//         Serial.println("ok");
if (Spos == 30) {
penDown();
}
if (Spos == 50) {
penUp();
}
break;
}
case 114:                              // M114 - Repport position
Serial.print( "Absolute position : X = " );
Serial.print( actuatorPos.x );
Serial.print( "  -  Y = " );
Serial.println( actuatorPos.y );
```

```
break;
default:
Serial.print( "Command not recognized : M");
Serial.println( buffer );
}
}
}
}
/*********************************
* Draw a line from (x0;y0) to (x1;y1).
* int (x1;y1) : Starting coordinates
* int (x2;y2) : Ending coordinates
*********************************/
void drawLine(float x1, float y1) {
if (verbose)
{
Serial.print("fx1, fy1: ");
Serial.print(x1);
Serial.print(",");
Serial.print(y1);
Serial.println("");
}
//  Bring instructions within limits
if (x1 >= Xmax) {
x1 = Xmax;
}
if (x1 <= Xmin) {
x1 = Xmin;
}
```

```
if (y1 >= Ymax) {
y1 = Ymax;
}
if (y1 <= Ymin) {
y1 = Ymin;
}
if (verbose)
{
Serial.print("Xpos, Ypos: ");
Serial.print(Xpos);
Serial.print(",");
Serial.print(Ypos);
Serial.println("");
}
if (verbose)
{
Serial.print("x1, y1: ");
Serial.print(x1);
Serial.print(",");
Serial.print(y1);
Serial.println("");
}
//  Convert coordinates to steps
x1 = (int)(x1*StepsPerMillimeterX);
y1 = (int)(y1*StepsPerMillimeterY);
float x0 = Xpos;
float y0 = Ypos;
//  Let's find out the change for the coordinates
long dx = abs(x1-x0);
```

```
long dy = abs(y1-y0);
int sx = x0<x1 ? StepInc : -StepInc;
int sy = y0<y1 ? StepInc : -StepInc;
long i;
long over = 0;
if (dx > dy) {
for (i=0; i<dx; ++i) {
myStepperX.onestep(sx,STEP);
over+=dy;
if (over>=dx) {
over-=dx;
myStepperY.onestep(sy,STEP);
}
delay(StepDelay);
}
}
else {
for (i=0; i<dy; ++i) {
myStepperY.onestep(sy,STEP);
over+=dx;
if (over>=dy) {
over-=dy;
myStepperX.onestep(sx,STEP);
}
delay(StepDelay);
}
}
if (verbose)
{
```

```
Serial.print("dx, dy:");
Serial.print(dx);
Serial.print(",");
Serial.print(dy);
Serial.println("");
}
if (verbose)
{
Serial.print("Going to (");
Serial.print(x0);
Serial.print(",");
Serial.print(y0);
Serial.println(")");
}

//  Delay before any next lines are submitted
delay(LineDelay);
//  Update the positions
Xpos = x1;
Ypos = y1;
}
//  Raises pen
void penUp() {
penServo.write(penZUp);
delay(penDelay);
Zpos=Zmax;
digitalWrite(15, LOW);
digitalWrite(16, HIGH);
if (verbose) {
```

```
Serial.println("Pen up!");
}
}
//  Lowers pen
void penDown() {
penServo.write(penZDown);
delay(penDelay);
Zpos=Zmin;
digitalWrite(15, HIGH);
digitalWrite(16, LOW);
if (verbose) {
Serial.println("Pen down.");
}
}
```

4.1.2 PROCESS 3.2.1 in JAVA CODE

```
import java.util.Date;

import java.awt.event.KeyEvent;

import javax.swing.JOptionPane;

import processing.serial.*;

import java.sql.*;

import java.io.*;

Serial port = null;

// select and modify the appropriate line for your operating system

// leave as null to use interactive port (press 'p' in the program)

String portname = null;

//String portname = Serial.list()[0]; // Mac OS X

//String portname = "/dev/ttyUSB0"; // Linux

//String portname = "COM6"; // Windows

boolean streaming = false;

float speed = 0.001;

String[] gcode;

int i = 0;

void openSerialPort()

{

if (portname == null) return;

if (port != null) port.stop();

port = new Serial(this, portname, 9600);
```

```
port.bufferUntil('\n');
}
void selectSerialPort()
{
String result = (String) JOptionPane.showInputDialog(frame,
"Select the serial port that corresponds to your Arduino board.",
"Select serial port",
JOptionPane.QUESTION_MESSAGE,
null,
Serial.list(),
0);
if (result != null) {
portname = result;
openSerialPort();
}
}
void setup()
{
size(600, 400);
openSerialPort();
}
void draw()
{
```

```
background(155);
fill(0);
int y = 24, dy = 12;
text("INSTRUCTIONS", 12, y); y += dy;
text("p: select serial port", 12, y); y += dy;
text("1: set speed to 0.001 inches (1 mil) per jog", 12, y); y += dy;
text("2: set speed to 0.010 inches (10 mil) per jog", 12, y); y += dy;
text("3: set speed to 0.100 inches (100 mil) per jog", 12, y); y += dy;
text("arrow keys: jog in x-y plane", 12, y); y += dy;
text("page up & page down: jog in z axis", 12, y); y += dy;
text("$: display grbl settings", 12, y); y+= dy;
text("h: go home", 12, y); y += dy;
text("0: zero machine (set home to the current location)", 12, y); y += dy;
text("g: stream a g-code file", 12, y); y += dy;
text("x: stop streaming g-code (this is NOT immediate)", 12, y); y += dy;
y = height - dy;
text("current jog speed: " + speed + " inches per step", 12, y); y -= dy;
text("current serial port: " + portname, 12, y); y -= dy;
}
void keyPressed()
{
if (key == '1') speed = 0.001; // set speed for motor
if (key == '2') speed = 0.01; // set speed for motor
```

```
if (key == '3') speed = 0.1; // set speed for motor

if (!streaming) {

if (keyCode == LEFT) port.write("G91\nG20\nG00 X-" + speed + " Y0.000 Z0.000\n"); //to machine go for left  side

if (keyCode == RIGHT) port.write("G91\nG20\nG00 X" + speed + " Y0.000 Z0.000\n"); //to machine go for right side

if (keyCode == UP) port.write("G91\nG20\nG00 X0.000 Y" + speed + " Z0.000\n");     //to machine check the up side

if (keyCode == DOWN) port.write("G91\nG20\nG00 X0.000 Y-" + speed + " Z0.000\n"); // to machine check the down side

if (keyCode == KeyEvent.VK_PAGE_UP) port.write("G91\nG20\nG00 X0.000 Y0.000 Z" + speed + "\n");

if (keyCode == KeyEvent.VK_PAGE_DOWN) port.write("G91\nG20\nG00 X0.000 Y0.000 Z-" + speed + "\n");

if (key == 'h') port.write("G90\nG20\nG00 X0.000 Y0.000 Z0.000\n"); // machine come to home

if (key == 'v') port.write("$0=75\n$1=74\n$2=75\n");

//if (key == 'v') port.write("$0=100\n$1=74\n$2=75\n");

if (key == 's') port.write("$3=10\n");

if (key == 'e') port.write("$16=1\n");

if (key == 'd') port.write("$16=0\n");

if (key == '0') openSerialPort();

if (key == 'p') selectSerialPort();// select the port

if (key == '$') port.write("$$\n");

}
```

```
if (!streaming && key == 'g')
{
gcode = null; i = 0;
File file = null;
println("Loading file...");
selectInput("Select a file to process:", "fileSelected", file);
}
if (key == 'x') streaming = false;
}
void fileSelected(File selection) {
if (selection == null) {
println("Window was closed or the user hit cancel.");
} else {
println("User selected " + selection.getAbsolutePath());
gcode = loadStrings(selection.getAbsolutePath());
if (gcode == null) return;
streaming = true;
System.out.println("harsh 2:"+selection.getAbsolutePath());
connection(selection.getAbsolutePath());
stream();
}
}
void connection(String filee)
```

```
{
Date d =new Date();
try
{
File f=new File(filee);
FileInputStream fis=new FileInputStream(f); //A FileInputStream obtains input bytes from a file in a file system
DataInputStream dis=new DataInputStream(fis); //application to read primitive data
System.out.println(dis+"");
Class.forName("com.mysql.cj.jdbc.Driver");
Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/harsh?useSSL=false","root","mh04bj4807");
System.out.println("done: "+con);
PreparedStatement ps=con.prepareStatement("insert into cnc(file, f_name, f_date) values(?,?,?)"); //file name and date will be show
System.out.println(""+ps);
ps.setBlob(1,fis);
ps.setString(2, f.getName());
ps.setString(3, d.toString());
int id=ps.executeUpdate();
if(id==1)
{
System.out.println("successful");
```

```
}

}

catch(Exception e)

{

System.out.println("hello"+e);

}

}

void stream()

{

if (!streaming) return;

while (true)

{

if (i == gcode.length) {

streaming = false;

return;

}

if (gcode[i].trim().length() == 0) i++;

else break;

}

println(gcode[i]);

port.write(gcode[i] + '\n');

i++;

}
```

```
void serialEvent(Serial p)

{

String s = p.readStringUntil('\n');

println(s.trim());

if (s.trim().startsWith("ok")) stream();

if (s.trim().startsWith("error")) stream(); // XXX: really?

}
```

4.1.3 G-CODE

(unicorn.py --tab="plotter_setup" --pen-up-angle=50 --pen-down-angle=30 --start-delay=150 --stop-delay=150 --xy-feedrate=3500 --z-feedrate=150 --z-height=0 --finished-height=0 --register-pen=true --x-home=0 --y-home=0 --num-copies=1 --continuous=false --pause-on-layer-change=false C:\Users\CG-DTE\AppData\Local\Temp\ink_ext_XXXXXX.svgEJS43Y)

G21 (metric ftw)

G90 (absolute mode)

G92 X0.00 Y0.00 Z0.00 (you are here)

M300 S30 (pen down)

G4 P150 (wait 150ms)

M300 S50 (pen up)

G4 P150 (wait 150ms)

M18 (disengage drives)

M01 (Was registration test successful?)

M17 (engage drives if YES, and continue)

(Polyline consisting of 2 segments.)

G1 X17.99 Y21.71 F3500.00

M300 S30.00 (pen down)

G4 P150 (wait 150ms)

G1 X16.30 Y23.83 F3500.00

G1 X14.51 Y25.01 F3500.00

G1 X13.47 Y25.17 F3500.00

G1 X12.56 Y24.98 F3500.00

G1 X11.50 Y23.97 F3500.00

G1 X10.96 Y24.15 F3500.00

G1 X8.75 Y25.19 F3500.00

G1 X7.56 Y25.29 F3500.00

G1 X6.52 Y25.03 F3500.00

G1 X5.71 Y24.47 F3500.00

G1 X5.49 Y23.70 F3500.00

G1 X6.09 Y22.10 F3500.00

G1 X6.41 Y21.57 F3500.00

G1 X3.32 Y22.61 F3500.00
G1 X1.07 Y23.96 F3500.00
G1 X-0.56 Y25.50 F3500.00
G1 X-1.54 Y27.20 F3500.00
G1 X-1.87 Y29.05 F3500.00
G1 X-1.53 Y30.72 F3500.00
G1 X-0.56 Y32.33 F3500.00
G1 X0.97 Y33.81 F3500.00
G1 X3.00 Y35.08 F3500.00
G1 X5.60 Y36.14 F3500.00
G1 X8.48 Y36.92 F3500.00
G1 X9.37 Y37.05 F3500.00
G1 X9.14 Y36.68 F3500.00
G1 X8.62 Y35.96 F3500.00
G1 X8.47 Y35.18 F3500.00
G1 X8.61 Y34.46 F3500.00
G1 X9.13 Y33.79 F3500.00
G1 X11.01 Y32.66 F3500.00
G1 X12.61 Y32.43 F3500.00
G1 X14.06 Y32.60 F3500.00
G1 X14.89 Y33.01 F3500.00
G1 X15.20 Y34.00 F3500.00
G1 X15.64 Y36.37 F3500.00
G1 X15.92 Y37.89 F3500.00
G1 X16.85 Y36.86 F3500.00
G1 X17.29 Y36.30 F3500.00
G1 X18.28 Y36.30 F3500.00
G1 X19.27 Y36.30 F3500.00
G1 X19.79 Y36.97 F3500.00
G1 X20.42 Y37.78 F3500.00
G1 X20.60 Y37.87 F3500.00
G1 X20.69 Y37.67 F3500.00
G1 X21.49 Y33.33 F3500.00
G1 X22.01 Y32.79 F3500.00

G1 X23.57 Y32.42 F3500.00
G1 X25.20 Y32.57 F3500.00
G1 X26.69 Y33.19 F3500.00
G1 X27.79 Y34.23 F3500.00
G1 X28.05 Y35.08 F3500.00
G1 X27.89 Y36.00 F3500.00
G1 X27.36 Y36.73 F3500.00
G1 X27.16 Y37.06 F3500.00
G1 X28.04 Y36.92 F3500.00
G1 X31.58 Y35.90 F3500.00
G1 X34.58 Y34.49 F3500.00
G1 X36.73 Y32.74 F3500.00
G1 X38.05 Y30.72 F3500.00
G1 X38.36 Y28.75 F3500.00
G1 X38.15 Y27.46 F3500.00
G1 X37.59 Y26.24 F3500.00
G1 X36.56 Y24.96 F3500.00
G1 X35.17 Y23.75 F3500.00
G1 X32.70 Y22.40 F3500.00
G1 X30.36 Y21.56 F3500.00
G1 X30.12 Y21.59 F3500.00
G1 X30.46 Y22.15 F3500.00
G1 X31.06 Y23.63 F3500.00
G1 X30.57 Y24.69 F3500.00
G1 X28.98 Y25.28 F3500.00
G1 X27.00 Y24.94 F3500.00
G1 X25.41 Y24.03 F3500.00
G1 X25.16 Y23.83 F3500.00
G1 X24.91 Y24.17 F3500.00
G1 X23.68 Y25.07 F3500.00
G1 X22.07 Y25.02 F3500.00
G1 X20.38 Y23.97 F3500.00
G1 X18.71 Y21.94 F3500.00
G1 X18.26 Y21.33 F3500.00

G1 X17.99 Y21.71 F3500.00

G1 X17.99 Y21.71 F3500.00

M300 S50.00 (pen up)

G4 P150 (wait 150ms)

(Polyline consisting of 2 segments.)

G1 X19.21 Y23.14 F3500.00

M300 S30.00 (pen down)

G4 P150 (wait 150ms)

G1 X21.05 Y24.86 F3500.00

G1 X22.04 Y25.33 F3500.00

G1 X23.17 Y25.43 F3500.00

G1 X24.10 Y25.26 F3500.00

G1 X24.84 Y24.65 F3500.00

G1 X25.20 Y24.27 F3500.00

G1 X25.68 Y24.58 F3500.00

G1 X27.54 Y25.44 F3500.00

G1 X28.87 Y25.56 F3500.00

G1 X30.14 Y25.30 F3500.00

G1 X31.27 Y24.28 F3500.00

G1 X31.31 Y23.35 F3500.00

G1 X30.91 Y22.28 F3500.00

G1 X30.81 Y21.99 F3500.00

G1 X31.69 Y22.28 F3500.00

G1 X34.37 Y23.60 F3500.00

G1 X36.46 Y25.24 F3500.00

G1 X37.42 Y26.55 F3500.00

G1 X37.99 Y27.99 F3500.00

G1 X38.08 Y29.26 F3500.00

G1 X37.82 Y30.51 F3500.00

G1 X36.67 Y32.36 F3500.00

G1 X34.81 Y33.98 F3500.00

G1 X32.26 Y35.32 F3500.00

G1 X29.07 Y36.37 F3500.00

G1 X27.93 Y36.65 F3500.00

G1 X28.04 Y36.39 F3500.00
G1 X28.33 Y35.04 F3500.00
G1 X28.14 Y34.23 F3500.00
G1 X27.48 Y33.46 F3500.00
G1 X26.25 Y32.63 F3500.00
G1 X24.80 Y32.19 F3500.00
G1 X23.28 Y32.14 F3500.00
G1 X21.86 Y32.52 F3500.00
G1 X21.32 Y32.88 F3500.00
G1 X20.64 Y36.17 F3500.00
G1 X20.45 Y37.32 F3500.00
G1 X20.13 Y36.92 F3500.00
G1 X19.62 Y36.27 F3500.00
G1 X19.43 Y36.02 F3500.00
G1 X18.30 Y36.02 F3500.00
G1 X17.16 Y36.02 F3500.00
G1 X16.99 Y36.21 F3500.00
G1 X16.47 Y36.85 F3500.00
G1 X16.13 Y37.30 F3500.00
G1 X16.08 Y37.10 F3500.00
G1 X15.71 Y35.15 F3500.00
G1 X15.33 Y33.15 F3500.00
G1 X15.01 Y32.71 F3500.00
G1 X14.27 Y32.35 F3500.00
G1 X12.61 Y32.16 F3500.00
G1 X10.90 Y32.38 F3500.00
G1 X8.92 Y33.58 F3500.00
G1 X8.33 Y34.33 F3500.00
G1 X8.19 Y35.19 F3500.00
G1 X8.41 Y36.22 F3500.00
G1 X8.58 Y36.65 F3500.00
G1 X7.46 Y36.37 F3500.00
G1 X4.79 Y35.53 F3500.00
G1 X2.55 Y34.49 F3500.00

G1 X0.75 Y33.26 F3500.00
G1 X-0.57 Y31.86 F3500.00
G1 X-1.49 Y29.89 F3500.00
G1 X-1.58 Y28.77 F3500.00
G1 X-1.35 Y27.56 F3500.00
G1 X-0.22 Y25.56 F3500.00
G1 X1.78 Y23.82 F3500.00
G1 X5.71 Y21.98 F3500.00
G1 X5.66 Y22.18 F3500.00
G1 X5.23 Y23.43 F3500.00
G1 X5.36 Y24.49 F3500.00
G1 X5.97 Y25.09 F3500.00
G1 X6.87 Y25.48 F3500.00
G1 X8.91 Y25.47 F3500.00
G1 X10.92 Y24.55 F3500.00
G1 X11.33 Y24.27 F3500.00
G1 X11.66 Y24.64 F3500.00
G1 X12.41 Y25.22 F3500.00
G1 X13.48 Y25.44 F3500.00
G1 X15.14 Y25.08 F3500.00
G1 X16.71 Y23.85 F3500.00
G1 X18.16 Y22.03 F3500.00
G1 X18.27 Y21.83 F3500.00
G1 X18.53 Y22.22 F3500.00
G1 X19.21 Y23.14 F3500.00
G1 X19.21 Y23.14 F3500.00
M300 S50.00 (pen up)
G4 P150 (wait 150ms)
(end of print job)
M300 S50.00 (pen up)
G4 P150 (wait 150ms)
M300 S255 (turn off servo)
G1 X0 Y0 F3500.00
G1 Z0.00 F150.00 (go up to finished level)

G1 X0.00 Y0.00 F3500.00 (go home)
M18 (drives off)

It’s a coordinate to drawing into paper and it’s change every time as per the user guideline

Install the Library and Drivers

- Adafruit library install into arduino its support for working stepper motor
- JDBC driver to connect an interface between java code and machine to save the record into SQL workbench
- inscape-unicorn-master library file into installation folder and the any image file save into G-code file to CNC machine work according the coordinate

4.2 TESTING APPROACH

4.2.1 Units Testing

Sr.No.	Test Cases	Conditions	Expected Output	Pass or fail
1	Arduino	Compile and upload the code	Uploading done and Successfully compiled	PASS
2	Inscape software	Create a vector into g-code	Successfully cratered g-code file with pen angle and buffer rate	PASS
3	Process3.2.1 software	The connectivity with system with machine to pass g-code	Successfully port selection and the upload the code to the machine	PASS
4	SQL Workbench	Save the record to see the how many time use the same data	Successfully entry in the table with real time with date	PASS
5	Steeper motor X,Y	Clockwise and antilock rotation	Successfully rotated with high-speed and low-speed	PASS
6	Servo motor	upwards and downwards movement	Successfully movement with accurate angel	PASS

Table .3 Units Testing

4.2.2 Integrated Testing

The integrated testing we see the start to end process the with detail study

- Create the image into g code with set the paper size and save the file
- Java code will we connect the system and machine and select the port and the connections will be done they give the message at the box and start the working
- The every g-code file will be save the details with real time and date we will see in the table
- After complete the work machine will go to the home position

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Test Reports

The Test Report we will see the set the jog speed with 3 mode and they will be work as per selection, all the components will work fine and accurate other test will be work fine as see in units testing and integrated testing

5.2 User Documentation

1. Start with the inscape software first set the paper size than add the image to the blank page after that click the path and trace bitmap then edge detection update and ok then delete the original image and set the bit map image go to save the image into g-code file at any place
2. Set up the machine with connect the cable and power supply
3. Open the process 3.2.1 software and run the java code popup the window to select the port and jog speed then press g and select the you created g-code file and then machine will be start the working
4. If you want to see the how many file machine will be work and then go to SQL workbench and the enter query and see the table with details of time and date

5.2.1 Screen shots

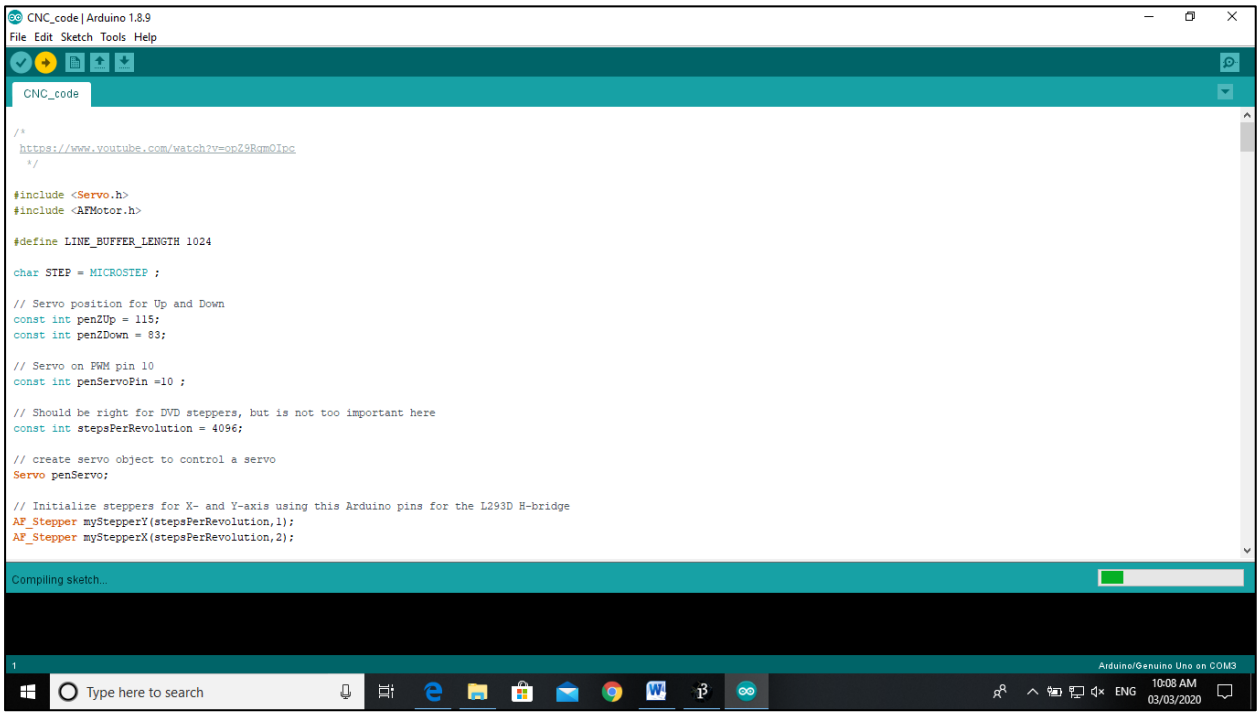

Fig 30 Arduino CNC coding and uploading

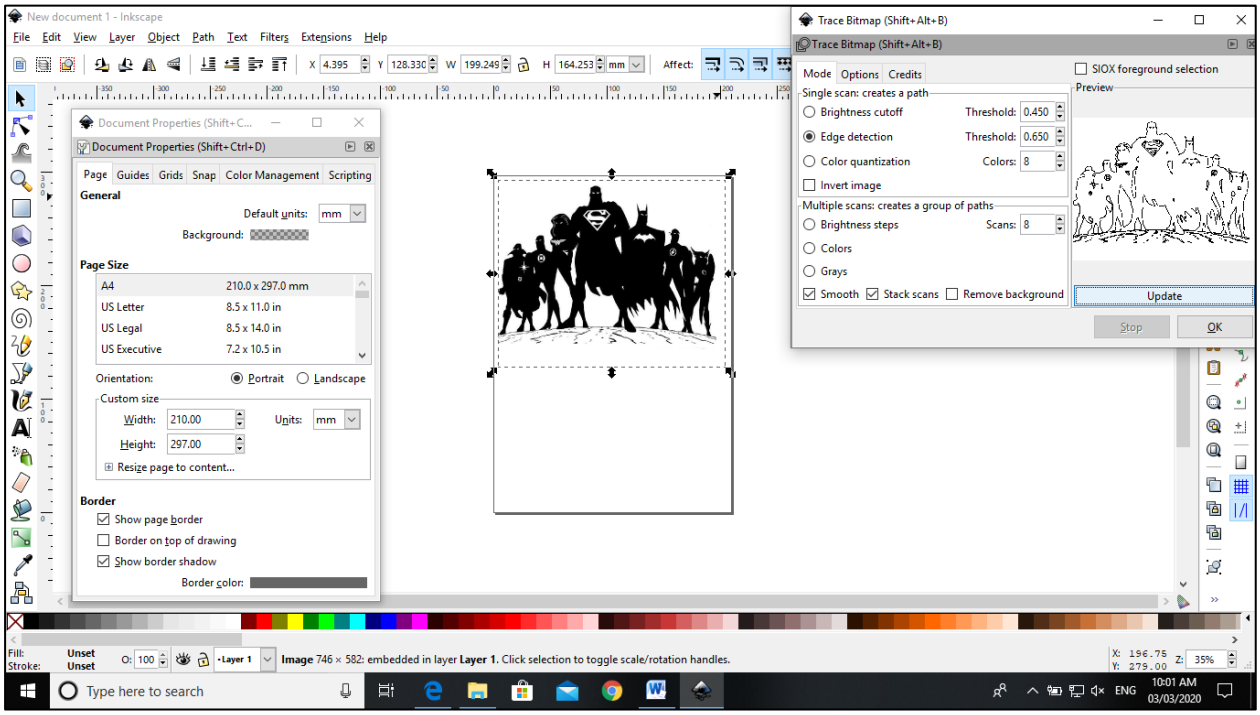

Fig 31 Inscape bitmap tracing

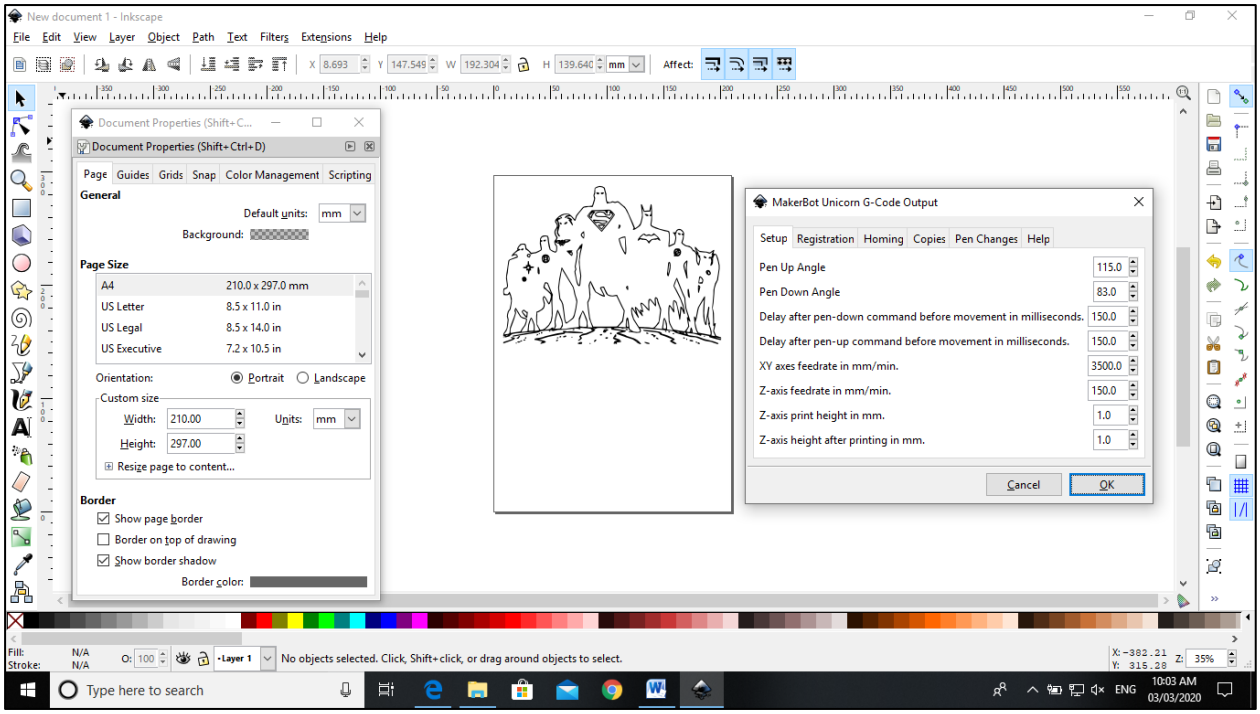

Fig 32 Save as the g-code and adjust the pen alignments

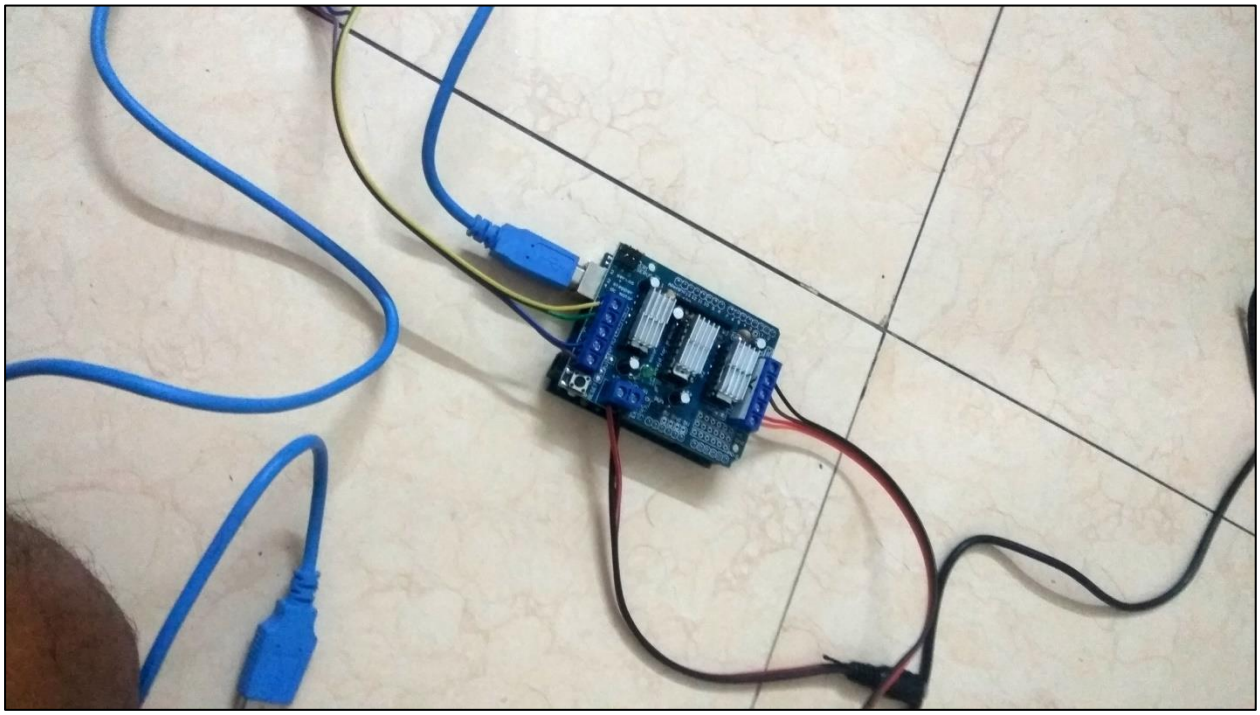

Fig 33 Connection with machine

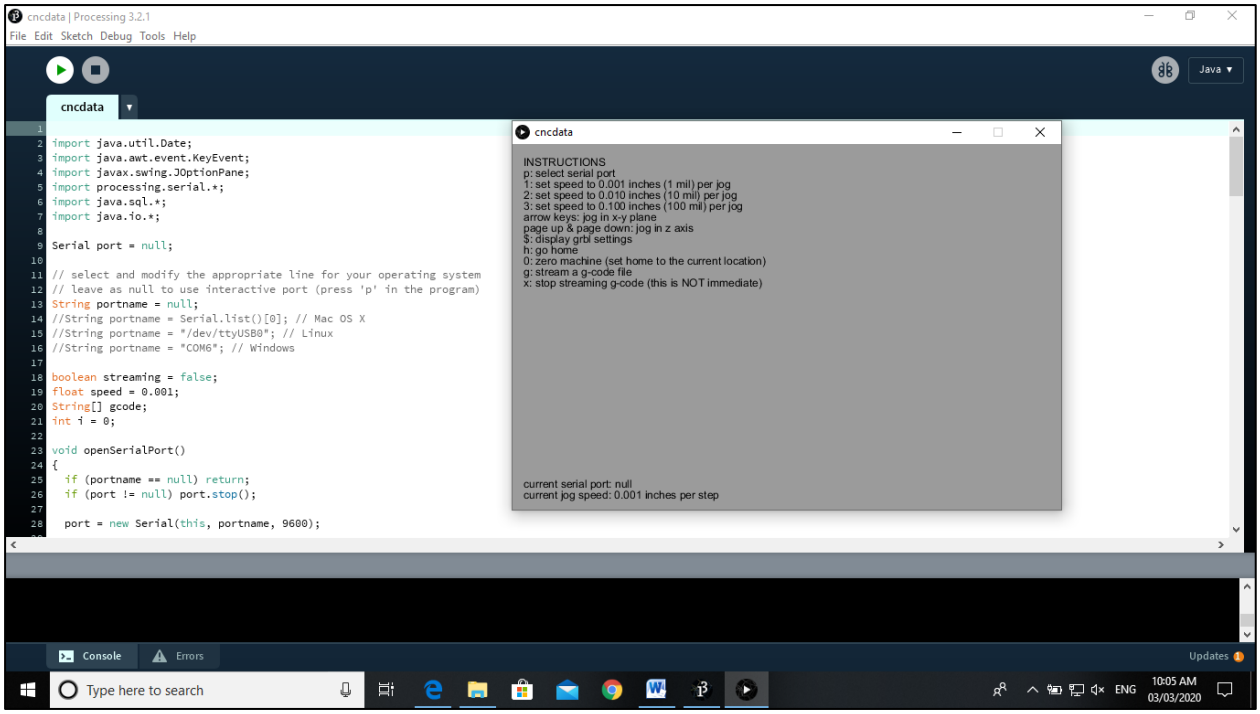

Fig 34 Run the java code and the selection of the port

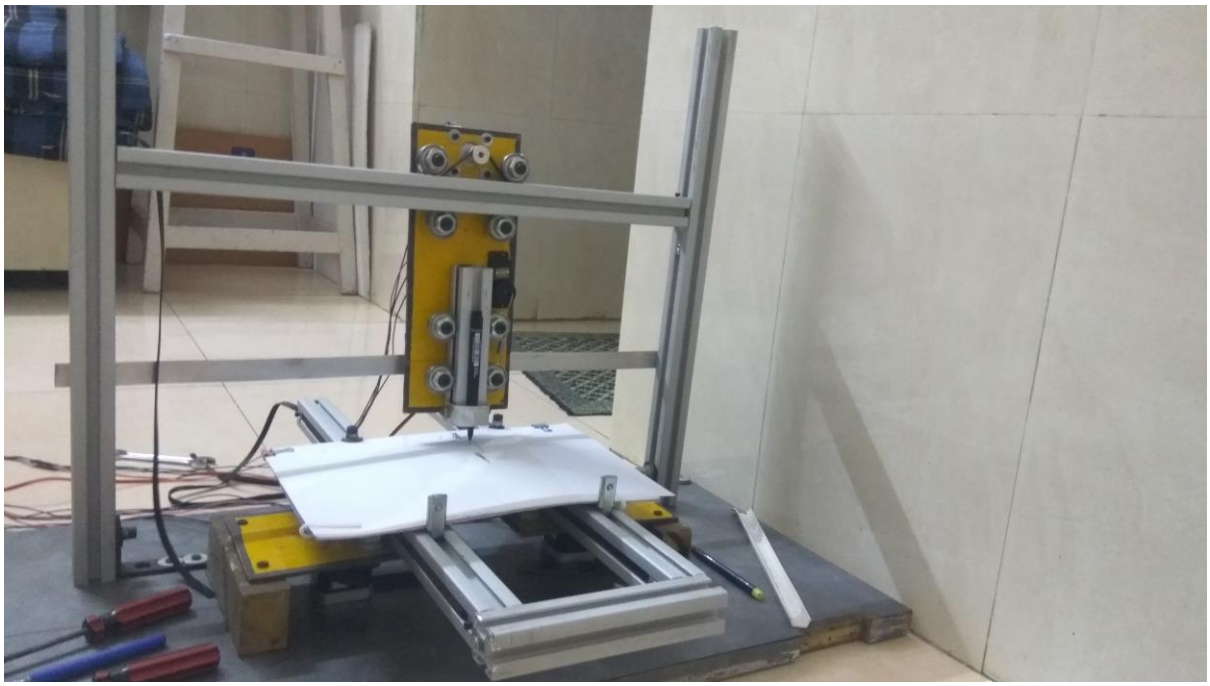

Fig 35 Machine start working

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

Hence I have successfully studied about CNC machine and their function. With their G code and operation performance on these type of function tool and many research to help to create a project

These machines have played a real important role in bringing about education revolution and laid the foundation .but the bringing about of the new technology in the present era is very important. The day by day the improve the quality and the improve the of the product its reduces the writing work of the student

This project is about building a mechanical prototype of a CNC plotter machine which is able to draw a layout of 297mm by 210mm (or any image/text) on a given solid surface. It consumes slow power and works with high accuracy due to precise controlling of stepper motors. This is a low cost project as compared to other CNC product. It is made with easily available components and spare parts. It is designed for private manufacturing and small scale applications in educational institutes. The machine is designed with a very simple construction scheme and can be carried anywhere without much effort. The algorithm used is simple. The pen can be replaced with a pinhead or laser head or any other tool for different purpose of use. Software that has been used is open source and user- friendly

6.2 Limitations of the System

They are costly Trained operator is required to operate the machine In case of breakdown a highly skilled professional is required to solve the problem also the programing knowledge and computer skills dc current will we in the given limit otherwise circuit will burn this machine become students lazy that’s way education system not approve the this type of Project

6.3 Modifications and Improvements

- Project we see the CNC shield will not work proper and the motor will work but not stop the rotations the change with Motor shield l293d
- The coding section we will change the coordinate and the buffer speed and the time delay
- In then project we change the wooden frame to aluminium profile
- Circuit will be heat up very fast we add the 12volt fan to cool down the and avoid the burring the circuit
- All the circuit will be attach in the small plastic box avoid the any other damage
- All the wire will cover by the PVC slave to avoid the stretch wire or break

6.4 Future Scope

In future the machine will be upgrade to laser engraving and 3d printing and circuit will be upgrade to higher version add other components to higher class of the machine to use in industries level and the normal citizen can create the your own business will the low cost price and more profit the this machine can save the time of all task just like a engraving, cutting, homework, printing, moulding parts In future in the event that we ready to interface this machine to the quick composition speed like Xerox machine then it will be more successful and will have the capacity to more number of pages in a brief span.

CHAPTER 7

BIBLIOGRAPHY

- https://www.creativitybuzz.org/how-to-make-homework-writing-machine/

  (this link are use for the over view of the machine work)

- https://www.skyfilabs.com/project-ideas/homework-writing-machine

  (this link is use for the asking the cnc shield code working)

- https://create.arduino.cc/projecthub/aakash11/writing-machine-0910e3

  (This link is get an idea for the how code working and settings)

- https://in.pinterest.com/search/pins/?q=homewrok%20writing%20machine&rs=typed&term_meta[]=homewrok%7Ctyped&term_meta[]=writing%7Ctyped&term_meta[]=machine%7Ctyped

  (This link for idea of the frame and the joints and inner and outer diameter)

- https://www.hackster.io/jithesh-thulasidharan/3d-printer-cnc-sketcher-homework-writer-sticker-cutter-combo-e70280

  (This link is for the asking for motor shield working and heating issue)

- https://github.com/martymcguire/inkscape-unicorn

  (This link for the referee of the coding)

- https://circuitdigest.com/microcontroller-projects/arduino-cnc-machine-project-code

  (This link use for the circuit diagram)

- http://www.sica.in/SubCategoriesDesign.aspx?categoryID=2&hasTarget=False

  (This link for the order the profile and accessories)

- https://www.ijert.org/research/Automated-Writing-and-Drawing-Machine--IJERTCONV7IS06049.pdf

  (This link for reference for the documentation and diagrams)

- https://www.youtube.com/watch?v=4QqvYT7D-i4

  (This link use for how to use machine)

- https://mosafavi.iut.ac.ir/sites/mosafavi.iut.ac.ir/files/files_course/cnc_1_0.pdf

  (This link is use to reference for the documentation)

- https://www.youtube.com/watch?v=bbe56S_O-uI

  (This link is use for how to create a g code and pen alignment)