EFFICIENT LEATHER CUTTER

By

Memona Mubashir

Remsha Ahmad

Supervisor

Dr. Muhammad Imran

Submitted to the faculty of Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology,
in partial fulfillment for the requirements of B.E Degree in Electrical
(Telecommunication) Engineering
July, 2020

CERTIFICATE OF CORRECTIONS & APPROVAL

Certified that work contained in this thesis titled **“**Efficient Leather Cutter*”*, carried out by NC Memona Mubashir and Remsha Ahmad under the supervision of Assoc. HoD Dr. Muhammad Imran for partial fulfillment of Degree of Bachelors of Electrical Engineering, in Military College of Signals, National University of Sciences and Technology, Islamabad during the academic year 2019-2020 is correct and approved. The material that has been used from other sources it has been properly acknowledged / referred.

Approved by

Supervisor

Dr. Muhammad Imran

EE Department, MCS

Date: 06 July 2020

DECLARATION

No portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student
Memona Mubashir
Remsha Ahmad

Registration Number
194018
154382

Signature of Supervisor

Acknowledgements

All praises to Allah, the most omnipotent, the most gracious, the most merciful, the most beneficent and the most compassionate and His blessings for the completion of our project. We thank ALLAH almighty for all the opportunities, trials and strength that have been bestowed upon us to achieve what we've achieved today. My humblest gratitude to the holy Prophet Muhammad (Peace Be Upon Him) whose way of life has been a continuous torch of guidance for us.

We have no proper words to convey our deep gratitude and respect for our project supervisor, Dr. Muhammad Imran Assoc. HOD from MCS who also provide us moral support in addition to providing us with technical help. He has inspired us to achieve our goal through hard work.

Our sincere thanks must also go to the members of faculty and staff of MCS who guided and supported us. They are memorable not only for their prompt support but also for their knowledge insight, advice, training which facilitated us throughout the project.

We also deeply thank our parents and friends, for their unconditional trust, timely encouragement, financial help and endless patience.

Finally, special thanks to our Electrical department of Military College of Signals, NUST, without which not even a single thing would've been possible.

At the end, we would like to thank all our instructors for their guidance and moral support throughout the journey.

Dedicated to the Martyrs of APS Peshawar attack

Abstract

Leather industry is a huge part of Pakistan's economy and it is facing issues due to its less efficient cutting methods. The leather cutting machines already available in industries are large in size and lack modern technology which makes the leather cutting process time consuming and costly. In this FYP, an efficient leather cutting machine is proposed which is based on the concept that the processed image of available leather piece will be fed to CNC machine, which will generate G-code thus controlling the positioning the working of laser. Processing of image also includes the selection of cutting designs available. The advantage of our venture lies in its low cost, variety of designs available and time efficiency.

Table of Contents

List of Figures

CHAPTER 1: INTRODUCTION

"LASER" is an acronym for light intensification by invigorated discharge of radiation and it was found in 1960. A laser is made when the electrons in particles in unique glasses, gems, or gasses absorb energy from an electrical source or another laser and get “excited." The energized electrons move from a lower energy orbit to a higher energy orbit around the molecule's core (nucleus). When they come back to their original energy level, the electrons radiate photons (particles of light). These photons are all at the same wavelength and peaks of the light waves are all in lockstep. Laser light is unique as compared to normal light. First of all, its light contains one and only wavelength (one particular shading/color). The specific wavelength of light is dictated by the measure of energy discharged when the excited electron drops to a lower energy orbit. Secondly, laser light is directional so it stays centered for limitless separations.

Lasers have the ability to cut complicated shapes, drill small holes and smooth welds with fiber free borders and flat work surface. Due to minimum tool requirement, there is a rapid development of prototype. Laser cutters are used to cut full shape for use in product, sign and stencil applications and they can cut or etch different materials including ply, hardwoods, acrylics in a variety of sizes, finishes and thicknesses, anodized aluminum, paper & much more. The accuracy level of cutting by laser cutters is better than any other traditional cutting methods as we get smooth edges with this technique.

1.1 Overview

CNC machine is an acronym for computerized numerical control machine which includes programmable automation in which processes are controlled by Alpha Numeric code. CNC machining involves the use of computers to control machine tools like lathes, mills and grinders. CNC machines can manufacture complex curved geometries in 2D or 3D which is extremely expensive when done by mechanical means because it requires complex jigs to control the cutter motion. CNC machine can control attached machining components with high precision and repeatability which leads to high productivity.

The utilizations of CNC machine incorporate both for non-machine and machine device apparatus. In the machine apparatus region, CNC is commonly utilized for drill press, processing machine, crushing unit, laser, sheet-metal press working machine, tube bowing machine etc. In the non-machine apparatus region, CNC applications include electronic get together, arranging measuring machine, welding machines (circular segment and resistance), tape laying and fiber twisting machines for composites and so on.

1.2 Problem definition

1. Absence of advanced technology in leather industry.
2. Large amount of manual work is required.
3. High cost of production.
4. Time consumption process.
5. Limitation of leather designs for goods

1.3 Objective

The objective of this project is to make an economical laser cutting machine which works by maximum utilization of the leather with minimal time utilization. The main objective is to accurately cut the leather by the commands given to the driving circuits of motors of CNC machine and for that purpose blue laser is used to increase effectiveness. Also, patterns are optimized in this project to utilize maximum amount of leather available and obtain a variety of patterns.

1.4 Scope

1. CNC machine used is highly efficient and the whole process is less time consuming because computer will control the whole process. Also, CNC can control movements of parts in any direction efficiently.
2. The CNC machine can run at a high speed, and it needs less manual work thus minimizing the cost of labor.
3. The cutting of leather is précised and accurate due to the laser used.

Optimization of pattern gives a huge variety of designs available for leather goods.

CHAPTER 2: LITERATURE REVIEW

CNC machine is used to manufacture 2D/3D complex curved geometries with precision and high repeatability. The high precision is due to the use of computer system to control machine parts/tools. CNC can control both machine and non-machine apparatus including drills, laser, processing machine, welding machine, arrange measuring machine and so on. The advantages of CNC includes high accuracy, flexibility, best spinning and cutting speeds of parts, automated tools, accurate control of movement of parts in any direction, high speed, minimum manual work required, availability of variety and feedback system [1].

Laser cutting techniques are always preferable in industry due to its collection of large amount of energy into a small defined area. This specific feature of laser cutting give it advantages including high level precision and accuracy of cut line with clean edges, cutting of different material sheets with various thickness, and low cost tools [2].

CNC Laser machine will control the movement of laser head by giving instructions to the movement parts through computer in the form of G-code. This G code is basically a set of instructions composed of letters, numbers and symbols which controls the movement of machine tool. Machine control unit (the heart of a CNC system) will decode the generated G-code and generate axis motion commands. CNC has a slide table controlled in X and Y axes and a spindle running along Z axis, with which laser head is mounted. The power drivers used are usually the electrical ones which include stepper and servo motors, which are inexpensive and also exhibit favorable torque speed characteristics [3].

2.1 Existing Literature:

CNC cutter has a slide table which uses 2 linear axes of movements, X and Y axis. These movements are guided by G-code generated by computer and executed by CNC machine. G-code is basically Alpha Numeric programming language. The speed at which the machine control transmits commands to driver system is very critical and old machines are obviously less capable in this regard. Driver system consists of AC/DC motors (servo and stepper motors) and these motors are responsible for the movement of CNC tools/parts [4].

Laser cutting device has a cavity designed to provide a controlled environment and a new configuration is set up to provide gas to the controlled environment with the cavity while the laser beam is used to cut metals and a moving tool mounted in the settled edge is used to disperse the laser beam efficiently. The movement of laser beam is instructed by PC by the use of GBRL firmware. This firmware moves according to the instructions given by G code, generated by inkscape software. For material cutting, the adequate force of laser bar needs to fall on the surface of the material and a rational laser bar source is utilized, free from other wellspring of unsettling influence. These laser cutting machines are utilized as a part of modern territory, for large scale manufacturing [5].

The three types of laser castoff in laser cutting mechanism are solid state lasers, laser micro jet and gas laser. Solid state laser is made by doping a crystalline solid host with ions, which will form actual active medium once embedded in the host material. They are very simple to produce and have high strength, high thermal conductivity and low absorption but they are quite expensive. Laser micro jet is a water-plane impelled laser which includes a laser bar blown at surface of the work piece alongside low weight water plane. The benefit is no warmth influenced zone and low running expense. In gas lasers, a gas discharge is pumped electrically in the active medium [6]. Beam quality absorption of laser beam is affected by wavelength, wave mode, power intensity, polarization and beam quality of the laser. The cutting speed of laser head is influenced by power, laser wavelength, focal point position, thickness and material of work piece [7].

CNC laser cutters also use optical technique to keep the PC updated of the speed and movement of laser head and also the results obtained from its movement. PC can control the working of the CNC machine by generating G code executed by GBRL firmware of the CNC machine. G code is a set of instructions containing the information of the direction and speed of the parts of CNC machine. Inkscape software is used to generate the G code, which is an open source software used for variety of purposes including artistic and technical illustrations such as cartoons, clip art, logos, typography, diagramming, generating SVG file and flowcharting. The speed and movement of CNC machine depends on the stepper and servo motors used [8].

Typically CNC laser cutters are used for industrial purposes on a large scale, which require high speed and low torque motors contributing to the efficiency of the cutting process. Stepper motors are simple in construction, cost effective, and have excellent start-stop and reversing responses but their torque capacity is low as compared to DC motors and they have limited speed. Servo motors provide high intermittent torque, high speed, smooth rotation at lower speed and work very well for velocity control but they are expensive than stepper motors and require tuning of control loop parameters. Also, they’re not suitable for hazardous environments [9].

CHAPTER 3: DESIGN ANALYSIS

CNC laser cutters reduce the time, work and cost of overall cutting process done by old fashioned cutters in industry, which are huge in size and lack the capability of efficient process.

CNC laser cutters reduce the time, work and cost of overall cutting process done by old fashioned cutters in industry. Also, CNC cutters are portable devices, which can be used by a small group of people in any situation with easiness and efficiency.

CNC laser cutters can cut leather with accuracy and efficiency into variety of deigns with high precision also. The edges of designs are also very clean due to the laser technique used, which concentrate a huge amount of light intensity on the specified region to cut it cleanly and accurately.

Also, CNC cutters are portable devices, which can be used by a small group of people in any situation with easiness and efficiency.

CNC laser cutters can cut leather with accuracy and efficiency into variety of deigns with high precision also. The edges of designs are also very clean due to the laser technique used, which concentrate a huge amount of light intensity on the specified region to cut it cleanly and accurately.

In this section, the components used for leather cutter are explained in details and how they contribute towards the efficiency and affectivity of this cutter. Also, the final plotter design is also explained which is a CNC machine.

3.1 Design requirements

The components used in leather cutter are given below along with the functions performed by them.

3.1.1 Stepper motor:

A stepper motor is a dc motor which is used to change the angular position of a rotor by in various steps. It’s basically a pulse driven motor widely used due to its low cost and high holding torque but it cannot run at high speed. There are two types of stepper motors: permanent magnet stepper motor and variable reluctance stepper motor. Two stepper motors (Nema 17) motors are used for x and y axis.

Nema 17 stepper motor is used in this project for:

- **Positioning of laser**- as stepper motor moves in precise repeatable steps, so it's used in applications like precise positioning such as CNC, 3D printers, and X, Y plotters.
- **Speed control of laser**- precise incremental movements make stepper motor perfect for rotational speed control in automation process and robotics.
- **Low speed torque**- usually DC motors don't produce adequate torque at low speed, but stepper motor has an advantage in this regard as it can produce maximum at low speeds. This feature makes it an excellent choice for applications requiring low speed with high precision.

Figure 1: Stepper motor

3.1.2 Servo motor:

A servo motor is a special kind of electromagnetic device producing precise degree of rotation. They're also called controllers as they control the mechanical system and they're basically an AC or a DC motor with a position sensing device.

Micro servo motor SG90 is used in this project to control the speed, torque and direction of laser in addition to monitoring its velocity and position also, which overall contributes to the movement of laser head.

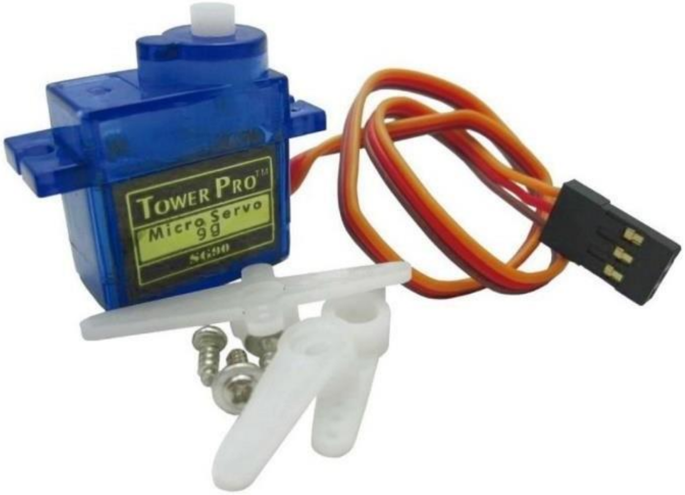


Figure 2: Servo motor

3.1.3 Blue Laser :

Blue laser emits electromagnetic radiation of 360-480 wavelengths, which will appear blue in color to a human naked eye. This laser will be used to cut leather in variety of different shapes/designs. The engaged laser head coordinated at the material, which then either softens, smolders, vaporizes away, or is passed up a plane of gas, leaving an edge with a top-notch surface completion. The focused laser beam first makes a melting zone on the surface of the given leather piece and when the laser head moves at a specific velocity, the cutting of leather begins.

3.1.4 CNC shield:

The CNC shield is an arduino compatible device which makes it possible to get your CNC extends up and running in a couple of hours. It uses an open source firmware to control stepper engines. CNC shield turns the arduino into CNC controller. CNC shield is inserted into Arduino UNO and install GBRL firmware is installed using Arduino IDE and then A4988 stepper motor drivers are inserted into cnc shield. Also, laser TTL pin is connected with z+ of CNC shield which is connected to pin 11 of Arduino.

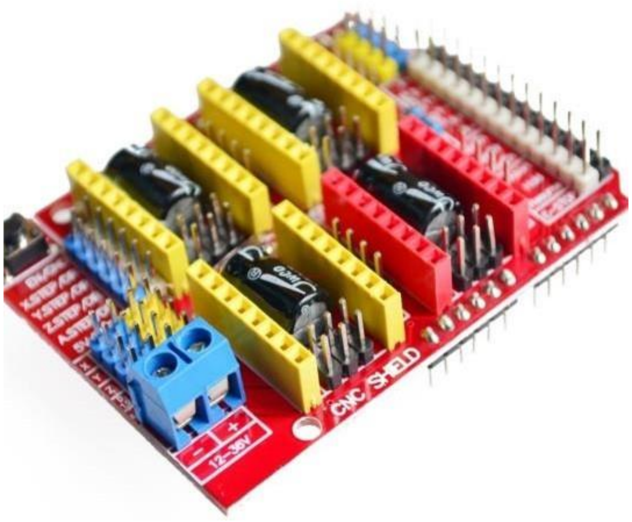


Figure 3: Arduino CNC shield

3.1.5 USB camera:

In this project, 10 Mega Pixel, high sensitivity USB camera is used, which will capture the image of the leather and then transmit it to computer for further image processing techniques. It has a 360 rotatable head and it can also capture high quality real time motional video. It's supported by windows XP/7/8/vista. The image captured by the camera is fed to the computer, which will process it in python for dimensions measurement and patterns optimization.

Figure 4: USB camera

3.2. Plotter design

3.2.1. 3D Model of Plotter

It’s a versatile machine which is used to serve a wide variety of everyday and specialized drawing and cutting purposes. It takes commands from computer in the form of GCODE to move the laser head, which will serve as a leather cutter. It will cut the pattern with such precision which is not even possible for a manual worker.

It can be used to cut any material, but here in this project it’s used to cut patterns on leather. The camera attached to the CNC machine capture the image of the leather placed on the surface and its dimensions are measured using different image processing software and techniques. Then PDF file will be converted to SVG file using INKSCAPE software which will then generate the G-code which will control the movement of CNC machine tools. This G-code will be send to GBRL firmware using INKSCAPE software which will control the servo and stepper motors, which will move the laser head to cut the leather placed on the surface.

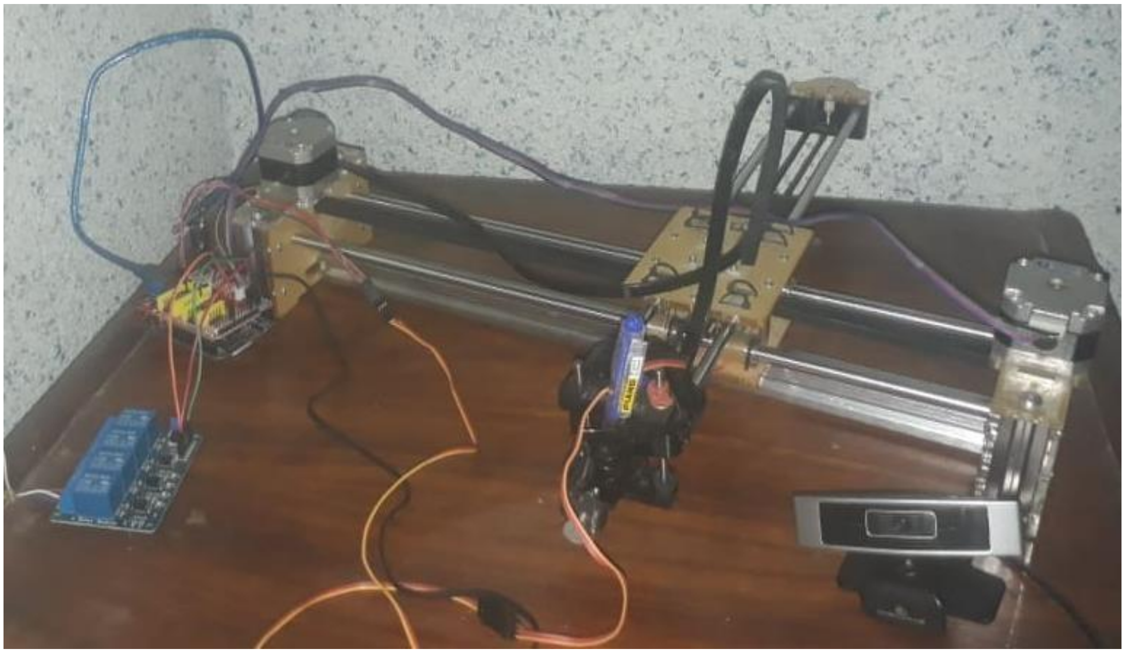


Figure 5: Plotter design

CHAPTER 4: PROPOSED WORK

The image of the work piece is captured by USB camera, and transmitted to PC, which measures its dimensions and optimized the patterns requested by the user with the help of Python, image processing software. The PDF file containing optimized patterns is converted to SVG file, portraying a vector art for the precision of designs using Inkscape software and after that G code is generated using that SVG file to guide the movement of laser head for cutting. The generated G code is then send to GBRL firmware using G code sender, which will control the speed, movement and direction of servo and stepper motors which in turn move the laser head.

Hardware and Software interface used in this project are given below:

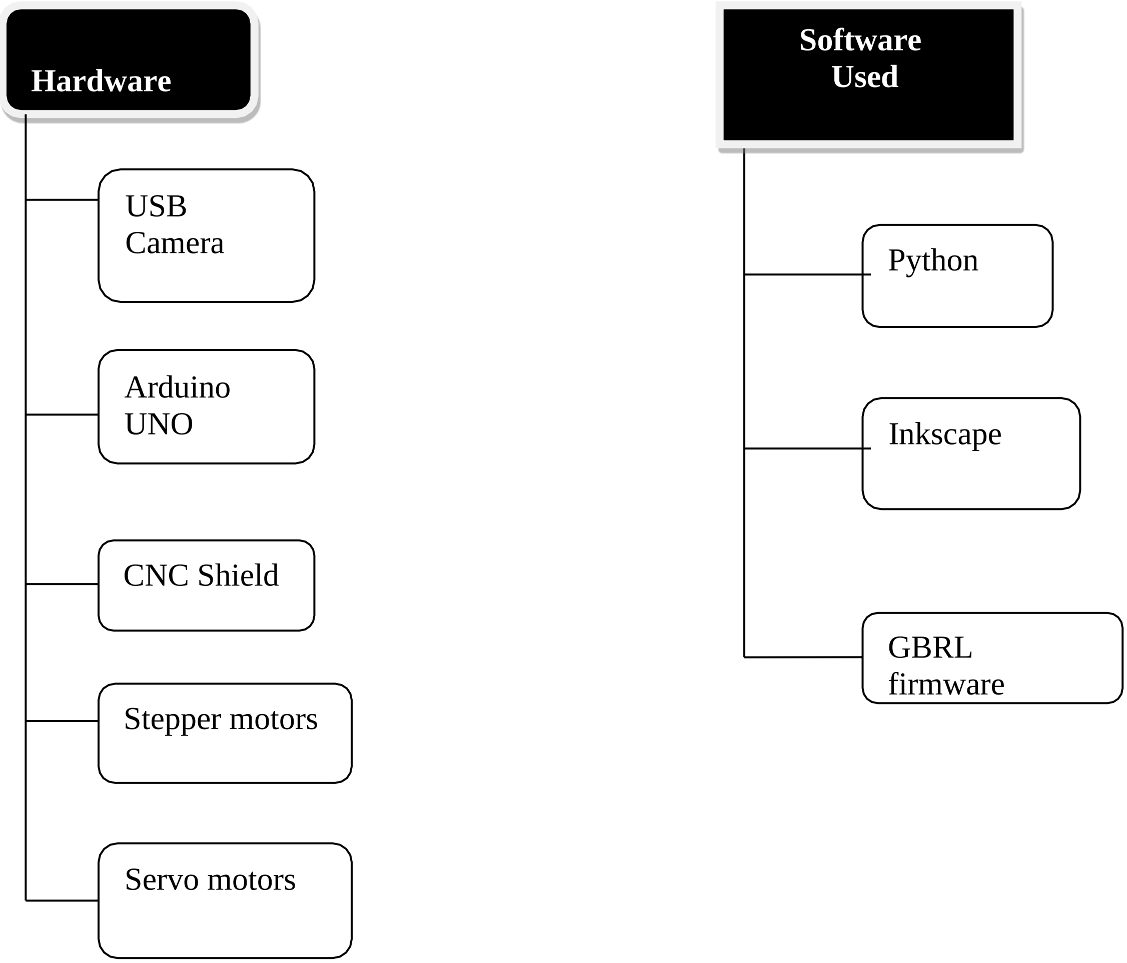


Figure 6: Hardware and Software interface used

The hardware and software approach used in the development of this project are explained in detail in this section.

4.1. Establishment

4.1.1. Description

The connection between application front end and firmware is of high priority as the working of whole system depends on it.

4.1.2. Response sequences

There should be a connection between application front end and firmware before the beginning of the process.

4.1.3. Requirements

The sub systems should be connected for the interlinked processes to begin.

4.2. Image capture of the leather

4.2.1. Description

USB camera will capture the image of the leather and send it to the attached computer. The image snapping of leather is controlled by the user.

4.2.2. Response sequences

The image captured by camera is transmitted to computer which will open it in Python for measuring the dimensions of leather and optimization of patterns.

4.2.3. Requirements

Connection between the camera and computer should be accurate and the position of the camera should be adjusted so that it can cover whole area of the leather.

4.3. Measuring dimensions of the leather

4.3.1. Description

The image captured by the camera will be opened in open CV, a library in Python which will measure the dimensions of the leather by pixel per metric ratio. To measure the size of the leather, we'll calibrate it with a reference object which should have two properties that is its dimension should be known and its placement should be identifiable.

4.3.2. Response sequences

Once the dimensions of the leather are known, the optimization of patterns will begin.

4.3.3. Requirements

To measure the pixel per metric ratio of the leather in open CV, the size of the reference object should be known and its position should be easily identifiable.

4.4. Optimum generation of patterns

4.4.1. Description

Optimization of patterns is done to maximize the utilization of available leather. User will enter the number of patterns needed and select the pattern designs and this data will be utilized by Python to arrange the patterns in such a way that the area they cover minimum area. This arrangement of patterns in done in such a way that they cover minimum convex Hull area and overlapping is zero.

4.4.2. Response sequences

Once the patterns are arranged, the PDF file will be generated and fed to inkscape software, which will be convert it into SVG file.

4.4.3. Requirements

The code should be properly working and the software should be efficient enough to generate the output as soon as the input is given.

4.5. Conversion of PDF into SVG file

4.5.1. Description

The PDF file will be given through application front end and accepted by the system for further processing.PDF file will be opened in inkscape software, which will convert it into SVG file and system will accept and successfully load it. If an error occurs, the system shall not proceed any further.

4.5.2. Response sequences

Loaded SVG file will be converted to G-code in the system for further processing.

4.5.3. Requirements

Successful loading of SVG file into the system is very important as well as the application front end should respond to show the loaded file.

4.6. Generation of G-CODE

4.6.1. Description

After the successful loading of SVG file in the system, it will convert into G-code object file for further processing. It is of high priority in our machine.

4.6.2. Response sequences

SVG file will be read and converted by the system into a G-code object file, which will instruct the system to proceed further with the cutting procedure.

4.6.3. Requirements

The SVG file should be converted into G-code type file precisely user should be able to view the final converted G-code before execution.

4.7. Sending G-CODE to GBRL

4.7.1. Description

After the error free conversion of SVG file into G-code type file, these instructions will be send to GBRL firmware for execution one by one and the cutting of the leather will begin.

4.7.2. Response sequences

System will transmitted from software one by one for execution upon the requests from firmware and the wait signal will be transmitted from firmware to software after the successful execution of the command.

4.7.3. Requirements

The transmission of commands between software and firmware should be smooth. Also, the system should stop immediately in case of any error to avoid damage in the final output and the status of pattern drawn should be shown at the application end to the user from the execution of very first command to the last one.

4.8. Laser cutting of selected pattern

4.8.1. Description

Laser head is controlled by firmware once the between system and application end is established.

4.8.2. Response sequences

Laser head will move according to the instructions given by G-code to draw the given pattern and It will stop once the pattern is drawn that is the last command is executed.

4.8.3. Requirements

The laser head should lift in case of idle situation and it is done from the application end and firmware once the connection between them is established.

4.9.Flow Chart

The whole process after establishing connection is explained previously. Now those steps are shown in the form of flow chart in this section. Image captured by the camera is processed in Open CV, a library of python and the final generated PDF file is converted into SVG file and G code is generated using this file in inkscape software. This G code instructs the motors and the motor drivers of CNC machine to move such that the laser head begins cutting of the work piece according to selected patterns.

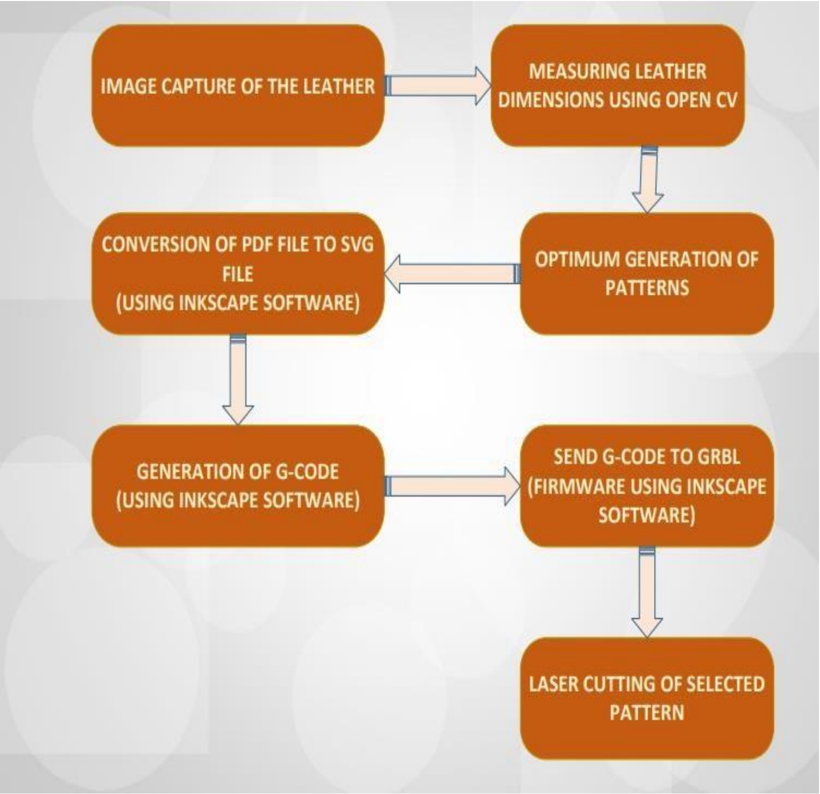


IMAGE CAPTURE OF THE LEATHER
MEASURING LEATHER DIMENSIONS USING OPEN CV
OPTIMUM GENERATION OF PATTERNS
CONVERSION OF PDF FILE TO SVG FILE (USING INKSCAPE SOFTWARE)
GENERATION OF G-CODE (USING INKSCAPE SOFTWARE)
SEND G-CODE TO GRBL (FIRMWARE USING INKSCAPE SOFTWARE)
LASER CUTTING OF SELECTED PATTERN

Figure 7: flow chart of the whole process

CHAPTER 5: SIMULATIONS AND RESULTS

Step 1:

An image of the given leather is captured using USB camera and transmitted to computer.

Step 2:

The image will be processed in Python to measure the dimensions of the given leather. It will be done according to code given in next section using the picture per metric ratio technique, as given by:

Pixels_per_metric=object_width/know_width

To measure the given object, it is calibrated using a reference object, whose dimensions and placement should be known.

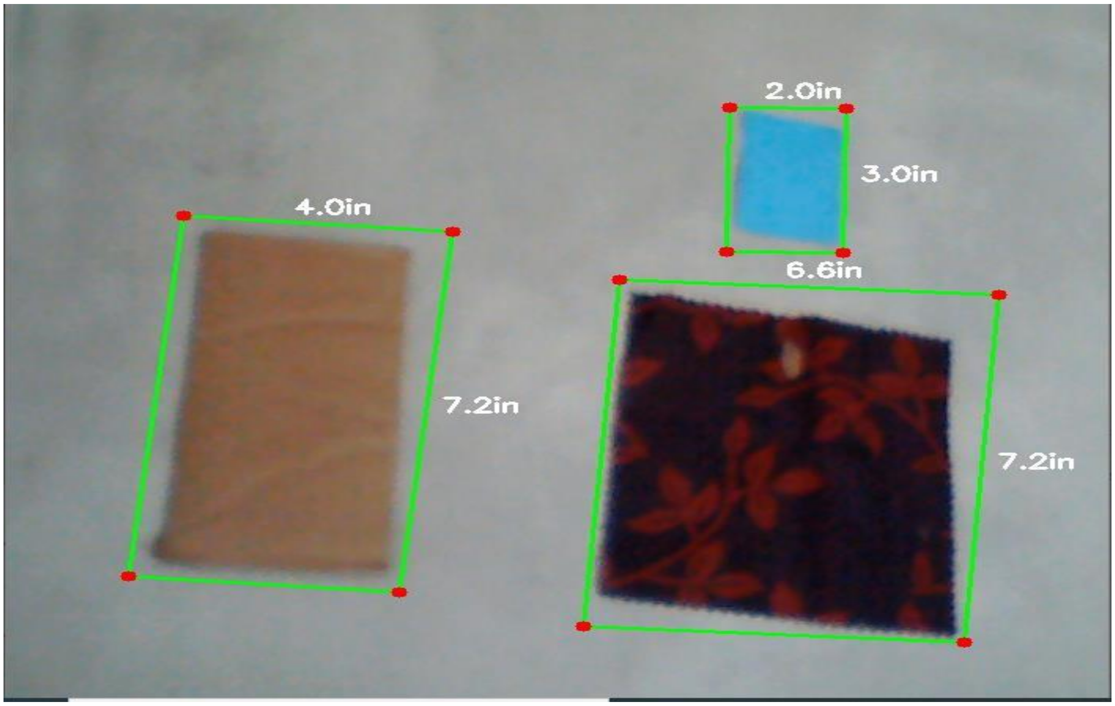


Figure 8: Measuring dimensions of objects

The table given below shows the dimensions of the objects measured using open CV software and the actual dimensions of the objects as measured physically. By comparing both observations, it can be seen that the dimensions measured using Python are not 100% accurate because the angle from ehich the image is taken is certainly not a perfect 90- degree angle. Also, the radial and tangential lens distortion also contributes towards this 1-2% error.

0BJECT	Measured width * length	ORIGINAL width * length	ACCURACY
1	4*7.2 inches	4*7.1 inches	99.286%
2	6.6*7.2 inches	6.8*7.1 inches	98%
3	2*3 inches	2.1*3 inches	97.6%

Figure 9: The measurements of the dimensions of objects

Step 3:

Further the number and type of patterns given by user will be optimized in the available area using convex hull method. Convex hull area is the minimum mean area consumed by the available shapes. The code is given in next section. The final PDF file generated with the maximum number of selected patterns optimized in the given work piece is shown in the given diagram.

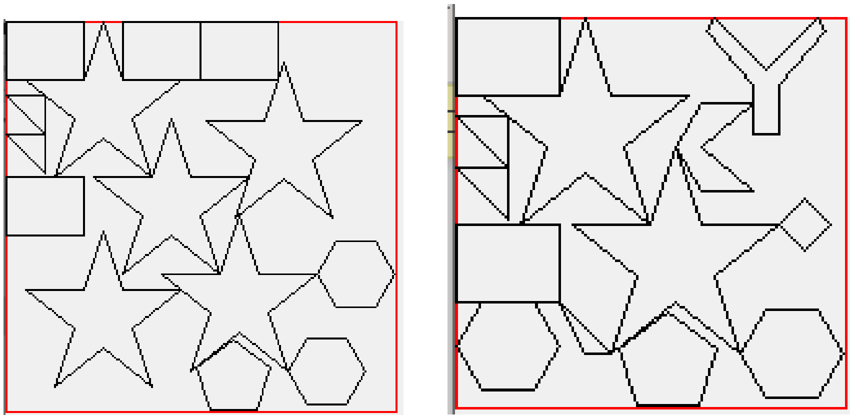
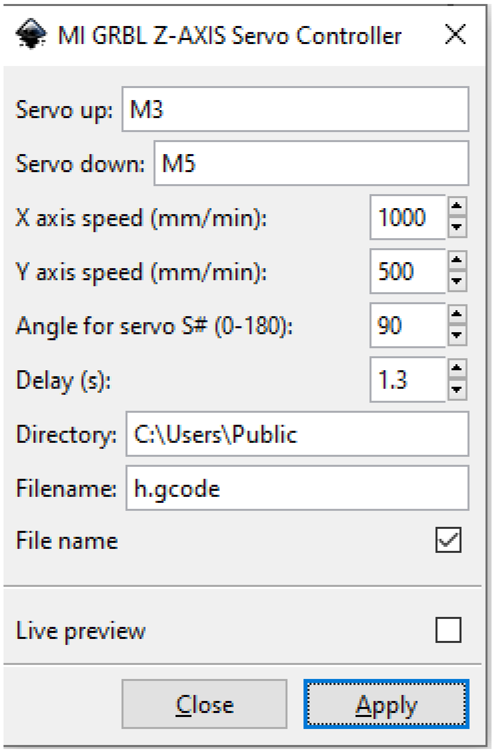


Figure 10: Optimized Patterns

Step 4:

Then the PDF file is converted into SVG file using inkscape software and also the G code is generated by using inkscape software by installing g code extension.



Servo up: M3

Servo down: M5

X axis speed (mm/min): 1000

Y axis speed (mm/min): 500

Angle for servo S# (0-180): 90

Delay (s): 1.3

Directory: C:\Users\Public

Filename: h.gcode

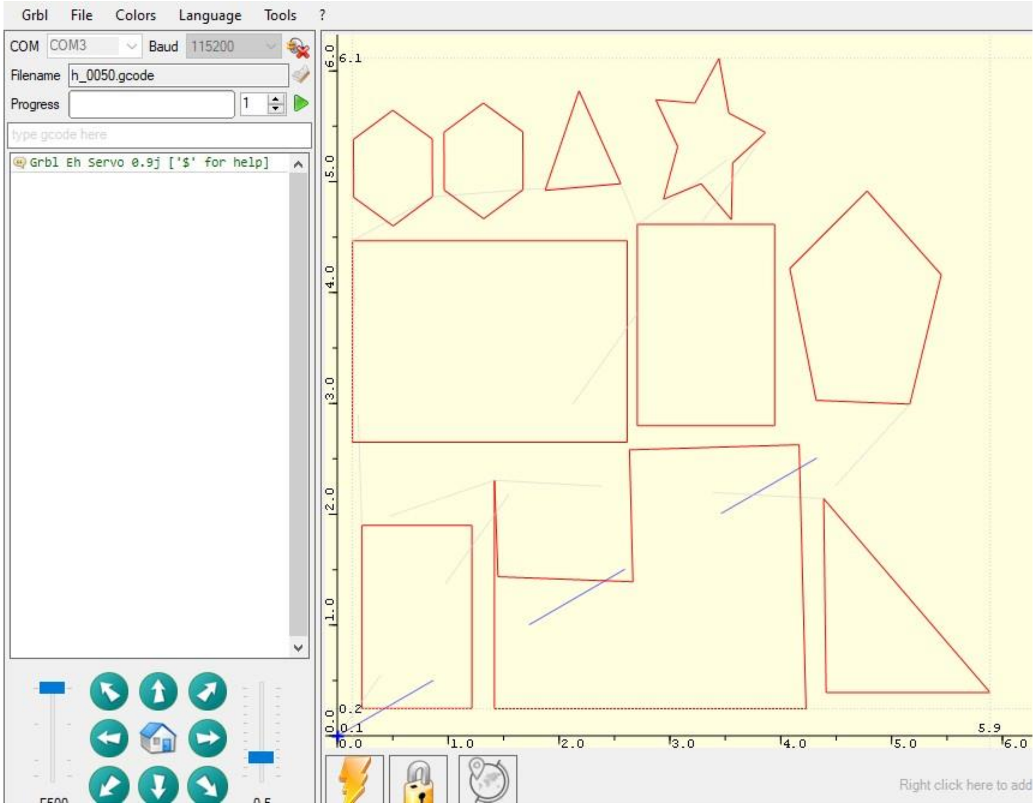
File name ☑

Live preview □

Figure 11: Generation of G-CODE

Step 5:

After that universal Gcode sender or laser GBRL is used to send Gcode to CNC machine (firmware) which will move laser in required direction and will cut the patterns. After running the G code, the resultant movement of laser head instructed by GBRL firmware is shown in the below diagram. It can be seen that the movement of the CNC machine head is according to the PDF file generated by python, the patterns are cut precisely and exactly like the ones generated by Python.



COM COM3 Baud 115200

Filename h_0050.gcode

Progress 1

Grbl Eh Servo 0.9j ['$' for help]

Figure 12: Simulation

Step 6:

The Arduino does not provide sufficient amount of output current to drive a motor. A stepper motor, which requires 350mA current for operation, if connected directly to the microcontroller will result in not working motor and destroying the microcontroller due to high current.

Due to these reasons, motor drivers are used which provide the sufficient amount of power required for motor operation.

The motor drivers are installed on the CNC shield which is then mounted on the arduino board. Make sure that the drivers are connected in the right direction**.**

STEP 7:

Now when the G code is sent to the arduino carrying GBRL library, the stepper signals are sent to the motors which draw their power from the drivers and move in the desired directions**.**

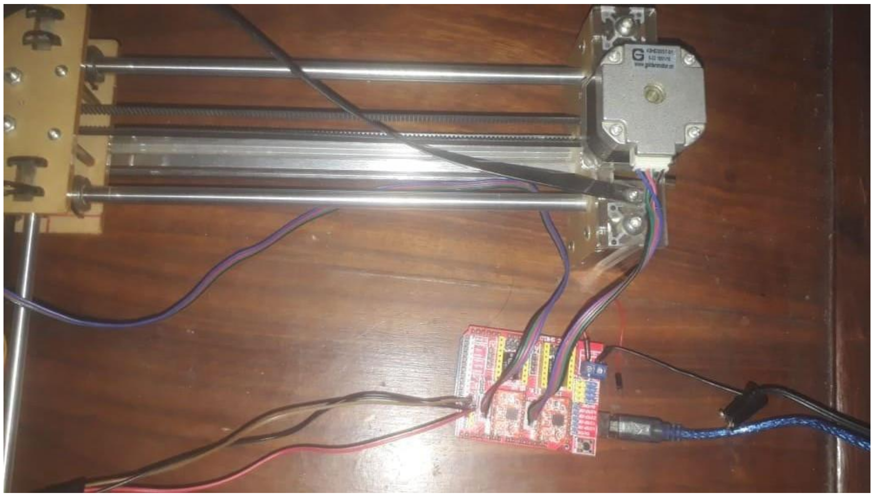


Figure 13: CNC and Arduino integration with motor

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1 Conclusions:

The simulation results show that this project effectively cut leather with minimum time and cost consumption. Also, the optimized patterns ensure the maximum utilization of work piece available.

6.2 Future work:

Laser cutter is a foundation stone towards revolutionizing the way materials are cut using laser. The future work mainly centers on the enhancement of the application that is user front end and the user transmission side and also on the type of cutters used for cutting different materials.

As a future improvement, a number of applications and techniques can be introduced due to the versatility and flexibility in its design.

1. A drill machine can be introduced in the place of laser, which can drill precisely in places required.
2. Different intensity lasers can be used to cut different materials like fabrics for a variety of designs.
3. It can be to draw different henna designs and also for drawing different models for industry purposes.

CHAPTER 7: CODE IMPLEMENTATION

7.1 Python code implementation

7.1.1. Measuring dimensions

% Importing packages and libraries

```
from scipy.spatial import distance as dist
from imutils import perspective
from imutils import contours
import numpy as np
import imutils
import cv2
```

% Midpoint of length and width of an object in an image

```
def midpoint(ptx, pty):
  return ((ptx[0] + pty[0]) * 0.5, (ptx[1] + pty[1]) * 0.5)
```

% Capturing frame

```
cap = cv2.VideoCapture(0)
while True:
   ret, frame = cap.read()
   cv2.imshow('frame',frame)
   key=cv2.waitKey(500)
   if key==27:
      break
```

% Performing canny edge detection and sorting contours

```
image= cv2.resize(frame,(700,700))
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray = cv2.GaussianBlur(gray, (5, 5), 0)
edged = cv2.Canny(gray, 10, 20)
```

```
        edged = cv2.dilate(edged, None, iterations=3)
        edged = cv2.erode(edged, None, iterations=1)
        cv2.imshow("edged",edged)

contours1 = cv2.findContours(edged.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
        contours1 = imutils.grab_contours(contours1)

        (contours1, _) = contours.sort_contours(contours1)
        pixelsPerMetric = None
        a=0
        listw=[]
        listl=[]
        pointx=[]
        pointy=[]
        point1x=[]
        point1y=[]
        listb=[]
        count=0
```

% Finding total number of objects

```
        cont=len(contours1)
        for c in contours1:
          if cv2.contourArea(c) < 700:
            continue
          orig = image.copy()
          count = count + 1
```

% Finding area of an object

```
        rect = cv2.minAreaRect(c)
        box  =  cv2.boxPoints(rect)
        box = np.int0(box)
box = perspective.order_points(box)
 listb.append(box)

        tl, tr, br, bl) = box
        (tltrX, tltrY) = midpoint(tl, tr)
        (blbrX, blbrY) = midpoint(bl, br)
        (tlblX, tlblY) = midpoint(tl, bl)
        (trbrX, trbrY) = midpoint(tr, br)

        dx = dist.euclidean((tl[0], tl[1]), (tr[0], tr[1]))
        dy = dist.euclidean((tl[0],tl[1]), (bl[0], bl[1]))
```

% Finding Pixel Per Metric Ratio

```
        if pixelsPerMetric is None:
          pixelsPerMetric = dx / 4
```

% Finding real dimensions of an object

```
        dimy = dy / pixelsPerMetric
        dimx = dx / pixelsPerMetric
        if(a==0) :
          dimx=dimx
          dimy=dimy-dimy*0.15
          a=1
        else :
            if (dimy < 3):
             dimy=dimy-0.13*dimy
```

```
                dimx=dimx+0.13*dimx
               if(dimy>3 and dimy<=8) :
                   dimy = dimy- dimy* 0.15
                   dimx = dimx + dimx * 0.27


        else:
                   dimy=dimy-dimy*0.11
                 dimx=dimx+dimx*0


listw.append(dimx) listl.append(dimy)
pointx.append(tltrX)
point1x.append(trbrX)
pointy.append(tltrY)
point1y.append(trbrY)
```

% Showing dimensions on an image

```
        for k in range (count):

          cv2.putText(orig, "{:.1f}in".format(listw[k]),

                (int(pointx[k] - 15), int(pointy[k] - 10)), cv2.FONT_HERSHEY_SIMPLEX,
                0.5, (255, 0, 255), 2)
          cv2.putText(orig, "{:.1f}in".format(listl[k]),

                (int(point1x[k] + 10), int(point1y[k])), cv2.FONT_HERSHEY_SIMPLEX,
                0.5, (255, 0, 255), 2)
          cv2.drawContours(orig, [listb[k].astype("int")], -1, (0, 255, 0), 2)
          for (x, y) in listb[k]:
            cv2.circle(orig, (int(x), int(y)), 5, (0, 0, 255), -1)
        cv2.imshow("Image", orig)
        cv2.waitKey(20000)
```

7.1.2. Optimization of patterns

% Importing packages and libraries

```
from graphics import *

from scipy.spatial import ConvexHull
area1=0

def  polynew(new):
    k= new.__len__()
    z = []

    for i in range(k):

        m = Point(new[i][0], new[i][1])
        z.append(m)
    new1 = Polygon(z)
    return (new1)
```

% Selection and customization of patterns

```
def create():

    option=input("enter for y already patterns and N for customization")
    if(option=='y'):
        r1 = [(0, 0), (40, 0), (40, 40), (0, 40)]

        r2 = [(0, 0), (40, 0), (40, 30), (0, 30)]

        r3 = [(0, 0), (10, 20), (20, 20)]

        r4 = [(0, 0), (20, 0), (20, 20)]

        r5 = [(60, 33), (40, 33), (30, 50), (40, 67), (60, 67), (40, 50)]

        r6 = [(20, 10), (30, 0), (40, 10), (30, 20)]

        r7 = [(20, 0), (40, 20), (60, 0), (63, 5), (45, 26), (45, 45), (35, 45), (35, 26), (17, 5)]
        r8 = [(50, 30), (31, 44), (38, 66), (62, 66), (69, 44)]

        r9 = [(30, 3), (10, 3), (0, 20), (10, 37), (30, 37), (40, 20)]
        r10=[(10,40),(40,40),(50,10),(60,40),(90,40),(65,60),(75,90),(50,70),(25,90),(35,60)]
        polygon = []
        polygon.append(r1)
        polygon.append(r2)
        polygon.append(r3)
        polygon.append(r4)
        polygon.append(r5)
        polygon.append(r6)
        polygon.append(r7)
        polygon.append(r8)
        polygon.append(r9)
        polygon.append(r10)
```

% User will enter the pattern number to select it

```
        choice = int(input("enter pattern number"))
        return (polygon[choice])
    else:
        lista = []
```

% For the customization enter the coordinates of pattern

```
        coor = int(input("number of coordinates"))
        for j in range(coor):
            x = int(input())
            y = int(input())
            m = (x, y)
            lista.append(m)


        return (lista)
```

% **Finding the coordinates of convex Hull**

```
def union(a,b) :


    from scipy.spatial import ConvexHull
    pt=[]
    convex=a+b
    hull=ConvexHull(convex)
    k=hull.vertices
    for i in range(k._len_()):
        pt.append(convex[k[i]]
           return (pt)
```

% Checking whether patterns are overlapping or not

```
def fun(r,s):

    from shapely.geometry import Polygon
    from shapely.ops import cascaded_union
    polygon1 = Polygon(r)
    polygon2 = Polygon(s)
```

% Using area technique to determine overlapping

```
    x=(polygon1.area +polygon2.area)
   # print (x)
    polygons = [polygon1, polygon2]
    u = cascaded_union(polygons)

    y=u.area
    #print(y)
    if(x==y) :
        return False
    else :
        return True
```

% Finding whether selected patterns lies within required area

```
def postcheck(new,length,width):
    count=0
    new1=polynew(new)
    k=new._len_()
    m=(new1.getPoints())

    for i in range(k) :
        x1 = m[i].getX()
        y1 = m[i].getY()
        if(x1<0 or x1>length or y1<0 or y1>width) :
            count = 1
            break


    if(count==1) :
        return False
    else :
        return True
```

% User will enter the length and width of a leather

```
length=int(input("enter the LENGTH"))
width=int(input("enter WIDTH"))
```

% Creating the window

```
win=GraphWin("h",700,700)
pat=int(input(" enter no of patterns "))
```

```
lista=[]
ci=[]
a1=[]
b1=[]
c1=[]
d1=[]
e1=[]
f1=[]
g1=[]
h1=[]
i1=[]
j1=[]
boundary=Polygon(Point(0,0),Point(length+1,0),Point(length+1,width+1),Point(0,width+1))
boundary.draw(win)

boundary.setOutline('red')
for i in range(pat):
```

% Asking for the selection of patterns

```
    lista=[]
    lista=create()

    ci=polynew(lista)
    if (i==0):
        a1=lista
        a2=ci
```

```
if (i==1):
    b1=lista
    b2=ci
if (i==2):
    c1=lista
    c2=ci
if (i==3):
    d1=lista
    d2=ci
if (i==4):
    e1=lista
    e2=ci
if (i == 5):
    f1 = lista
    f2 = ci
if (i == 6):
    g1 = lista
    g2 = ci
if (i == 7):
    h1 = lista
    h2 = ci
if (i == 8):
    i1 = lista
    i2 = ci
if (i == 9):
    j1 = lista
    j2 = ci
```

% fixing first patterns

```
for k in range (pat-1):
area1=0
    first=[]
    change=[]
    if (i == 7):
        h1 = lista
        h2 = ci
    if (i == 8):
        i1 = lista
        i2 = ci
    if (i == 9):
        j1 = lista
        j2 = ci
```

% Getting the vertices of a pattern

```
for i in range(a1.__len__()):
    x1 = (a2.getPoints())
    x1 = x1[i].getX()
    y1 = (a2.getPoints())
    y1 = y1[i].getY()
    for j in range(b1.__len__()):
        x2 = (b2.getPoints())
        x2 = x2[j].getX()
        y2 = (b2.getPoints())
        y2 = y2[j].getY()
        x = x1 - x2
        y = y1 - y2
        new = [(a + x, b + y) for a, b in b1]
```

% Placing second pattern on first patterns vertices

```
check=postcheck(new,length,width) if(check==True):
```

% Calling function to check overlapping and moving to next step

```
m = fun(a1, new)
if m == False:
convex = a1 + new
```

% Finding the convex shape of patterns

```
    hull = ConvexHull(convex)
    area = (hull.volume)
    print(area)
    if (area1 == 0):
        area1 = area
        poly1 = union(a1, new)
        first=new
```

% Finding the minimum area and exact position of a pattern

```
    else:

        if (area <= area1):
            area1 = area
            poly1 = union(a1, new)
            first=new
   else:
         print("overlapping")
         b1 = first
        b2 = polynew(first)
```

% Finding the exact position for all patterns , repeating the above process for all patterns by looping

```
    else:
      if (k == 1):
        test1 = c1
        test2 = c2
      if (k == 2):
        test1 = d1
        test2 = d2
      if (k == 3):
        test1 = e1
        test2 = e2
      if (k == 4):
        test1 = f1
        test2 = f2
      if (k == 5):
        test1 = g1
        test2 = g2
      if (k == 6):
        test1 = h1
        test2 = h2
      if (k == 7):
        test1 = i1
        test2 = i2
      if (k == 8):
        test1 = j1
        test2 = j2
```

```
for i in range(pol1.__len__()):
    x1 = (pol2.getPoints())
    x1 = x1[i].getX()
    y1 = (pol2.getPoints( ))
    y1 = y1[i].getY()
    for j in range(test1.__len__()):
        x2 = (test2.getPoints())
        x2 = x2[j].getX()
        y2 =(test2.getPoints())
        y2 = y2[j].getY()
        x = x1 - x2
        y = y1 - y2
        new = [(a + x, b + y) for a, b in test1]

        m = fun(pol1, new)
        check =postcheck(new,length,width)
        if (check == True):
            if m == False:
                convex = pol1 + new

                hull = ConvexHull(convex)
                area = (hull.volume)
                print(area)
                if (area1 == 0):
                    area1 = area

                    change = new
                    poly1 = union(pol1, new)
                else:
                    if (area <= area1):
                        area1 = area
                        change = new

                    poly1 = union(pol1, new)
                else:
                    print(" overlapping")
    if (k == 1):
     c1 = change
    c2=polynew(change)
    if (k == 2):
     d1 = change
    d2=polynew(change)
    if (k == 3):
     e1 = change
     e2 = polynew(change)
    if (k == 4):
     f1 = change
    f2=polynew(change)
    if (k == 5):
     g1 = change
    g2=polynew(change)
    if (k == 6):
     h1 = change
    h2=polynew(change)
    if (k == 7):
     i1 = change
    i2=polynew(change)
    if (k == 8):
     j1 = change
     j2=polynew(change)

poly2 = polynew(poly1)
pol1=poly1
pol2=poly2
```

% drawing the patterns according to new position.

```
for k in range(pat):
    if(k==0):
        a2.draw(win)
    if(k==1):
        b2.draw(win)
    if (k == 2):
        c2.draw(win)
    if (k == 3):
        d2.draw(win)
    if (k == 4):
        e2.draw(win)
    if (k == 5):
        f2.draw(win)
    if (k == 6):
        g2.draw(win)
    if (k == 7):
        h2.draw(win)
    if (k == 8):
        i2.draw(win)
    if (k == 9):
        j2.draw(win)
ext1=0
listb = []
listc = []
         newextra=[] new=[]
        extra=8
```

% Checking for extra space in a leather for more patterns

```
for ext in range (extra):
    let=0
    b = create()
    listb.append(b)
    put=0
    for x in range(length):
        listc.append( polynew(listb[ext]))
        for y in range(width):
            i = 0
            x1 = (listc[ext].getPoints())
            x1 = x1[0].getX()
            y1 = (listc[ext].getPoints())

            y1 = y1[0].getY()
            new =([(a + x, b + y) for a, b in listb[ext]])
```

% Look for space and place the patterns .

% Checking the overlapping and then finally placing new patterns.

% Repeat the same procedure for all patterns

% Repeat until no or very few area left

```
            if (pat==2):
                m = fun(new, a1)
                m1 = fun(new, b1)
                for h in range(ext1):


                    n=fun(newextra[h],new)
                    if(n==True):
                        i=1
```

```
    if (m == False and m1 == False and i == 0 ):
        check = postcheck(new, length, width)

        if (check == True and let==0):
            newextra.append(new)
            z = polynew(newextra[ext1])
            z.draw(win)
            let=1
             i = 1
            ext1=ext1+1
            put=1
if (pat == 3):
    m = fun(new, a1)
    m1 = fun(new, b1)
    m2=fun(new, c1)
    for h in range(ext1):
        n = fun(newextra[h], new)
        if (n == True):
            i = 1
    if (m == False and m1 == False and m2==False and i == 0):
        check = postcheck(new, length, width)

        if (check == True and let == 0):
            newextra.append(new)
            z = polynew(newextra[ext1])
            z.draw(win)
            let = 1
            i = 1
            ext1 = ext1 + 1
            put=1
if (pat == 4):
```

```
            if (pat == 4):
                        m = fun(new, a1) m1 = fun(new,

                        b1) m2 = fun(new, c1)

                m3=fun(new,d1)

                for h in range(ext1):

                    n = fun(newextra[h], new)

                    if (n == True):

                        i = 1
               if (m == False and m1 == False and m2 == False and m3==False and i == 0):

                    check = postcheck(new, length, width)

                    if (check == True and let == 0):

                        newextra.append(new)

                        z = polynew(newextra[ext1])

                        z.draw(win)

                        let = 1

                        i = 1

                        ext1 = ext1 + 1

                        put=1

            if (pat == 5):

                m = fun(new, a1)

                m1 = fun(new, b1)

                m2 = fun(new, c1)

                m3=fun(new,d1)

                m4 = fun(new, e1)

                for h in range(ext1):
                    n = fun(newextra[h], new)

                    if (n == True):

                        i = 1
                if (m == False and m1 == False and m2 == False and m3==False and

m4==False and i == 0):

                    if (check == True and let == 0):

                        newextra.append(new)
                        z = polynew(newextra[ext1])
                        z.draw(win)
                        let = 1
                        i = 1
                        ext1 = ext1 + 1
                        put=1
            if (pat == 6):
                m = fun(new, a1)
                m1 = fun(new, b1)
                m2 = fun(new, c1)
                m3 = fun(new, d1)
                m4 = fun(new, e1)
                m5 = fun(new, f1)
                for h in range(ext1):
                    n = fun(newextra[h], new)
                    if (n == True):
                        i = 1
                if (m == False and m1 == False and m2 == False and m3 == False and m4 ==
False and m5==False and i == 0):
                    check = postcheck(new, length, width)
                    if (check == True and let == 0):
                        newextra.append(new)
                        z = polynew(newextra[ext1])
                        z.draw(win)
                        let = 1
                        i = 1
                        ext1 = ext1 + 1
                        put=1
            if (pat == 7):
                m = fun(new, a1)
                m1 = fun(new, b1)
                m2  =  fun(new,  c1)
                m3  =  fun(new,  d1)
                m4  =  fun(new,  e1)
                m5  =  fun(new,  f1)
                m6  =  fun(new,  g1)
                for h in range(ext1):
                    n = fun(newextra[h], new)
                    if (n == True):
                        i = 1
                if (m == False and m1 == False and m2 == False and m3 == False and m4 ==
False and m5 == False and m6==False and i == 0):
                    check = postcheck(new, length, width)
                    if (check == True and let == 0):
                        newextra.append(new)
                        z = polynew(newextra[ext1])
                        z.draw(win)
                        let = 1
                        i = 1
                        ext1 = ext1 + 1
                        put=1
            if (pat == 8):
                m = fun(new, a1)
                m1 = fun(new, b1)
                m2 = fun(new, c1)
                m3 = fun(new, d1)
                m4 = fun(new, e1)
                m5 = fun(new, f1)
                m6 = fun(new, g1)
                m7 = fun(new, h1)
                for h in range(ext1):
```

```
                    n = fun(newextra[h], new)
                    if (n == True):
                        i = 1

                if (m == False and m1 == False and m2 == False and m3 == False and m4 ==
False and m5 == False and m6==False and m7==False and i == 0):
                    check = postcheck(new, length, width)
                    if (check == True and let == 0):
                        newextra.append(new)
                        z = polynew(newextra[ext1])
                        z.draw(win)
                        let = 1
                        i = 1
                        ext1 = ext1 + 1
                        put=1
            if (pat == 9):
                m = fun(new, a1)
                m1 = fun(new, b1)
                m2 = fun(new, c1)
                m3 = fun(new, d1)
                m4 = fun(new, e1)
                m5 = fun(new, f1)
                m6 = fun(new, g1)
                m7 = fun(new, h1)
                m8=fun(new, i1)
                for h in range(ext1):

                    n = fun(newextra[h], new)
                    if (n == True):
                        i = 1
                if (m == False and m1 == False and m2 == False and m3 == False and m4 ==
False and m5 == False and m6 == False and m7 == False and m8==0 and i == 0):
                    check = postcheck(new, length, width)
```

```
        if (check == True and let == 0):
            newextra.append(new)
            z = polynew(newextra[ext1])
            z.draw(win)
            let = 1
            i = 1
            ext1 = ext1 + 1
            put=1
if (pat == 10):
    m = fun(new, a1)
    m1 = fun(new, b1)
    m2 = fun(new, c1)
    m3 = fun(new, d1)
    m4 = fun(new, e1)
    m5 = fun(new, f1)
    m6 = fun(new, g1)
    m7 = fun(new, h1)
    m8= fun(new, i1)
    m9 = fun(new, j1)


    for h in range(ext1):
        n = fun(newextra[h], new)
        if (n == True):
            i = 1
    if (m == False and m1 == False and m2 == False and m3 == False and m4 == False and m5 == False and m6 == False and m7 == False and m8==0 and m9==0 and i == 0):
        check = postcheck(new, length, width)
        if (check == True and let == 0):
            newextra.append(new)
```

```
                z = polynew(newextra[ext1])
                z.draw(win)
                let = 1
                i = 1
                ext1 = ext1 + 1
                put=1
                if(put==1):
                break
win.getMouse()
```

7.2 Gcode implementation

% G code as generated by Inkscape software

% M5 and M3 for switching of Laser

% G commands to move stepper motors to specific coordinates.

```
M5
G90
G21
G1 F1000
G1 X5.1642 Y2.9932
M3 S90
G4 P1.2999999523162842
G1 F500.000000
G1 X4.3196 Y3.028
G1  X4.0813 Y4.2167
G1  X4.7786 Y4.9163
```

```
G1 X5.4479 Y4.16
G1 X5.1642 Y2.9932
M5
G4 P1.2999999523162842
G1 F1000
G1 X4.387 Y2.1404
M3 S90
G4 P1.2999999523162842
G1 F500.000000
G1  X4.4091 Y0.3923
G1  X5.8883 Y0.3923
G1 X4.387 Y2.1404
M5
G4 P1.2999999523162842
G1 F1000
G1 X1.4176 Y2.3064
M3 S90
G4 P1.2999999523162842
G1 F500.000000
G1  X1.4176 Y0.2462
G1  X4.2276 Y0.2462
G1  X4.1651 Y2.6269
G1  X2.6352 Y2.5813
G1  X2.6665 Y1.3909
```

```
G1  X1.4488 Y1.4365
G1  X1.4176 Y2.3064
M5
G4 P1.2999999523162842
G1 F1000
G1 X0.2204 Y1.8983
M3 S90
G4 P1.2999999523162842
G1 F500.000000
G1  X1.2139 Y1.8983
G1  X1.2139 Y0.2473
G1  X0.2204 Y0.2473
G1  X0.2204 Y1.8983
M5
G4 P1.2999999523162842
G1 F1000
G1 X0.1365 Y4.4668
M3 S90
G4 P1.2999999523162842
G1 F500.000000
G1  X2.6146 Y4.4668
G1  X2.6146 Y2.6499
G1  X0.1365 Y2.6499
G1  X0.1365 Y4.4668
```

```
M5
G4 P1.2999999523162842
G1 F1000
G1 X0.8575 Y4.8639
M3 S90
G4 P1.2999999523162842
G1 F500.000000
G1  X0.5031 Y4.6009
G1  X0.1469 Y4.8599
G1  X0.1451 Y5.3819
G1  X0.4995 Y5.6449
G1 X0.8557 Y5.386
G1 X0.8575 Y4.8639
M5
G4 P1.2999999523162842
G1 F1000
G1 X1.6744 Y4.9287
M3 S90
G4 P1.2999999523162842
G1 F500.000000
G1 X1.32 Y4.6657
G1  X0.9637 Y4.9246
G1  X0.9619 Y5.4467
G1  X1.3163 Y5.7097
```

```
84. G1 X1.6725 Y5.4507
85. G1 X1.6744 Y4.9287
86. M5
87. G4 P1.2999999523162842
88. G1 F1000
89. G1 X2.5568 Y4.9823
90. M3 S90
91. G4 P1.2999999523162842
92. G1 F500.000000
93. G1 X1.874 Y4.9222
94. G1  X2.1799 Y5.8194
95. G1  X2.5568 Y4.9823
96. M5
97. G4 P1.2999999523162842
98. G1 F1000
99. G1 X2.706 Y4.6166
100.    M3 S90
101.    G4 P1.2999999523162842
102.    G1 F500.000000
103.     G1  X3.9451 Y4.6166
104.     G1  X3.9451 Y2.7998
105.     G1  X2.706 Y2.7998
106.     G1  X2.706 Y4.6166
107.     M5
```

```
G4 P1.2999999523162842
G1 F1000
G1 X3.8636 Y5.4442
M3 S90
G4 P1.2999999523162842
G1 F500.000000
G1 X3.566 Y5.169
G1 X3.5541 Y4.6587
G1    X3.281 Y4.9827
G1    X2.9414 Y4.8414
G1    X3.0703 Y5.3168
G1    X2.8724 Y5.7398
G1    X3.2251 Y5.7097
G1    X3.4423 Y6.1124
G1    X3.5314 Y5.6184
G1    X3.8636 Y5.4442
M5
G4 P1.2999999523162842
G1 F1000
G1 X0 Y
```

REFERENCES:

1. A. M. Sheri, B. Ali, U. Naveed and Z. A. Abassi, "MR PEN", July 2017.
2. H. Ferdinando, I.N. Sanjaya, "Automatic Driling Machine for Printed Circuitr Board" Proceedings of the 6th Symposium on Advanced Intelligent Systems, Surabaya Indonesia, pp. 100-120, 2005.
3. N. Balasubramanyam and Prod. Smt.G. Prasanthi "Design and Fabrication of an Automatic PC- Based Drilling Machine", HTCL Open International journal of Engineering Technology innovations and Research, vol.7, pp. 80-92, January 2014.
4. D.S. Berstein, "Setting up and running a control research laboratory", IEEE Control Systems Magazine, vol. 23, pp.14-19, 200laboratory", Proc. India Int. Conf. Power Electronics, Chennai, pp. 289-292, 2006.
5. M. Ibrahim, S. Fayyaz, A. Tasleem and M. Haris, "Printed Circuit Board Plotter", June 2018.