Faculty: Artificial Intelligence Year/Level: Bio-3 Subject: Projects

Group Name:		Group Logo:
Error Team		

The Team:

مروة السيد خليل السيد

إيمان فتحي منصور فتحي

آيه سعد باب الفرج المحروق

لمياء احمد محمد عبد العزيز

محمود رفعت على كراوية

احمد كمال الدين حسين السنوسي

First project :

- **Pendulum Clock:**
- **We used C language to build a Pendulum Clock.**
- **We used geometric figures (circle, rectangle, arc, line) to build it.**
- **Common functions that we used:**
- **Circle (), rectangle (), setcolor (), outtextxy ()….etc.**

Output:

Code:

```
#include<conio.h>
#include<iostream.h>
#include<graphics.h>
#include<dos.h>
#include<math.h>
#include<stdlib.h>/*The stdlib.h header defines four variable types, several macros,
and various functions for performing general functions.*/
#include<alloc.h>
 void mai n()
  {
   int d=DETECT,m,r=50;
   float gx,gy,x,y,x1,y1,xm,ym,xh,yh,k=0,i,j,ii,l;
```

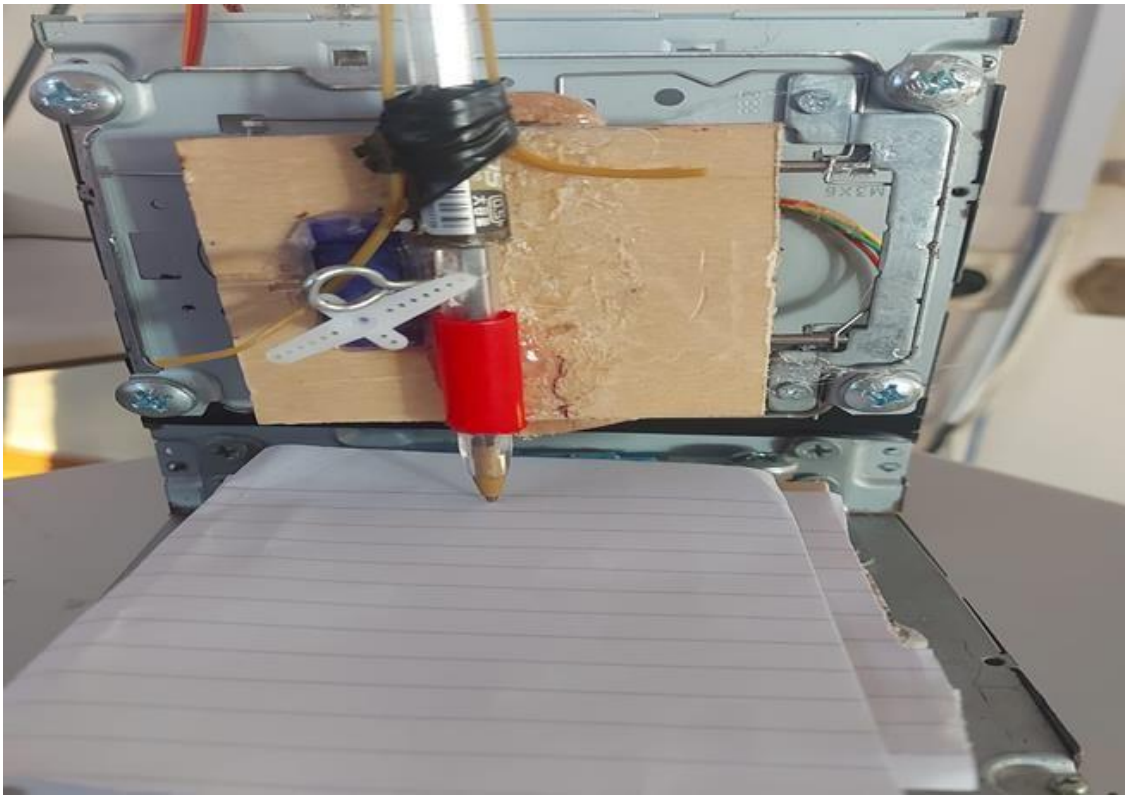
```
   initgraph(&d,&m,"");
   struct time t;//Structure containing a calendar date and time broken down into
its components.
   void *mem;//pointer
   int size;
  ii=13.89;
 setcolor(14);
 line(159,100,159,387);
 line(450,100,450,382);
 line(159,100,450,100);



 setcolor(14);
 arc(300,100,236,307,250);

 arc(300,100,244,298,320);


 while(1)
 {

 while(ii<14.36)

  {
  if(kbhit())//kbhit() is present in conio.h and used to determine if a key has been
pressed or not,If a key has been pressed then it returns a non zero value otherwise
returns zero.
     exit(1);
   // second's pixel
  for(i=11;i<=17.3;i+=0.105)
   {
    gx=55*cos(i);
    gy=55*sin(i);
    putpixel(gx+300,gy+200,4);

   }
   // outer circle
      setcolor(2);
       circle(300,200,85);
       circle(300,200,87);
       circle(300,200,90);
       setcolor(3);
       outtextxy(510,440,"Made By :-");
       outtextxy(530,460,"Hitesh Kumar");


      setcolor(14);
       circle(300,200,80);

      setcolor(4);
       outtextxy(330,143,"1");
       outtextxy(350,165,"2");
       outtextxy(360,197,"3");
       outtextxy(352,228,"4");
       outtextxy(328,252,"5");
       outtextxy(298,260,"6");
       outtextxy(264,252,"7");
```

```
    outtextxy(242,226,"8");
    outtextxy(235,197,"9");
    outtextxy(235,165,"10");
    outtextxy(260,140,"11");
    outtextxy(292,135,"12");
//time function
///*
    gettime(&t);
    gotoxy(35,20);
   cout<<int(t.ti_hour)<<":"<<int(t.ti_min)<<":"<<int(t.ti_sec)<<"";
   i=(int(t.ti_sec)*(0.105))+11;
   j=(int(t.ti_min)*(0.105))+11;
   k=((int(t.ti_hour)*(0.105))*5+11);

   int min=int (t.ti_min);
   int rem=min/12;
   k=k+(rem*0.105);
///*/
//calculations for second hand
   x=r*cos(i);
   y=r*sin(i);
   setcolor(14);
 line(300,200,x+300,y+200);

//calculations for minute hand
   xm=(r-5)*cos(j);
   ym=(r-5)*sin(j);
   setcolor(9);
 line(300,200,xm+300,ym+200);

 //calculations for hour hand
   xh=(r-20)*cos(k);
   yh=(r-20)*sin(k);
   setcolor(6);
 line(300,200,xh+300,yh+200);

  delay(200.5);

   int pks=0;


if(kbhit())
exit(1);
float l,m;
  for(pks=1;pks<=4;pks++)
  {

  l=200*cos(ii);
  m=200*sin(ii);

   setcolor(4);
  line(300,290,300+l,190+m);


   setcolor(14);
   setfillstyle(SOLID_FILL,14);
  circle(l+300,m+190,15);
   floodfill(l+300,m+190,14);
```

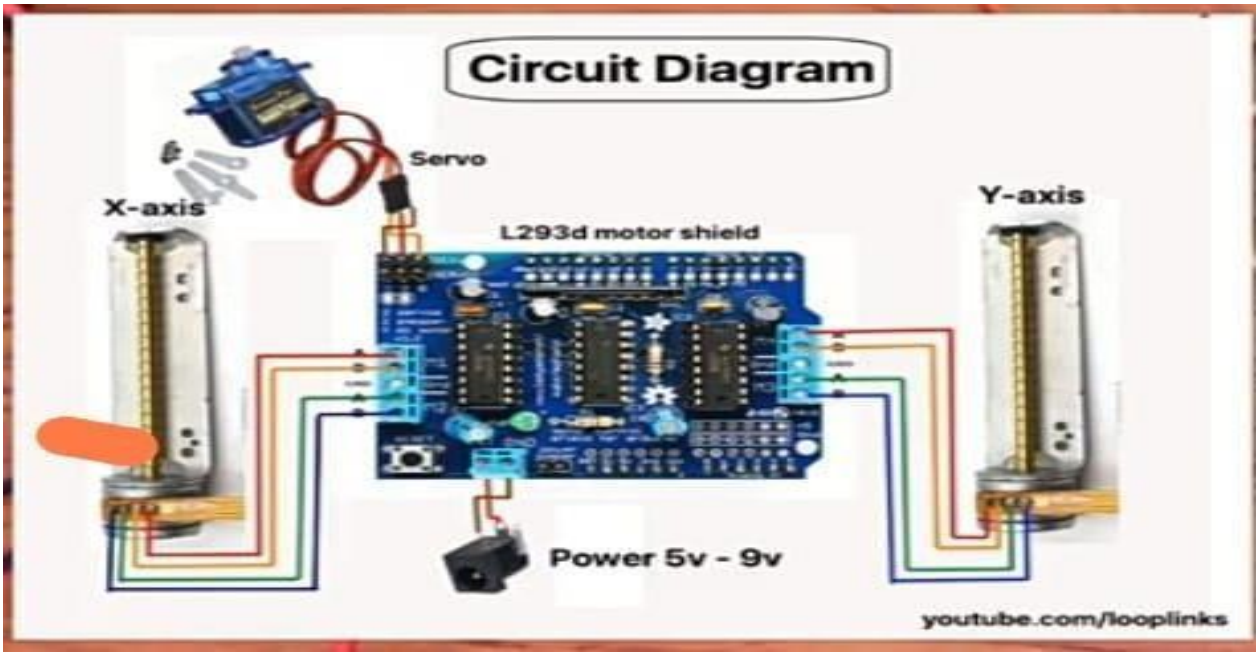
```
      ii+=.105;

    delay(200.5);
      setcolor(0);
      setfillstyle(SOLID_FILL,0);
     circle(l+300,m+190,15);
      floodfill(l+300,m+190,0);

      line(300,290,300+l,190+m);
    }

    setcolor(0);
line(300,200,x+300,y+200);
    line(300,200,xm+300,ym+200);
    line(300,200,xh+300,yh+200);
}

    while(ii>13.89)
    {
    if(kbhit())
     exit(1);

 for(i=11;i<=17.3;i+=0.105)
  {
   gx=55*cos(i);
   gy=55*sin(i);
   putpixel(gx+300,gy+200,4);
  }
      setcolor(12);
          circle(300,200,85);
          circle(300,200,87);
          circle(300,200,90);

        setcolor(4);
       outtextxy(510,440,"Made By :-");
       outtextxy(530,460,"Hitesh Kumar");

      setcolor(15);
    setfillstyle(SOLID_FILL,0);
     circle(300,200,80);
    floodfill(300,200,0);
     setcolor(4);
      outtextxy(330,143,"1");
      outtextxy(350,165,"2");
      outtextxy(360,197,"3");
      outtextxy(352,228,"4");
      outtextxy(328,252,"5");
      outtextxy(298,260,"6");
      outtextxy(264,252,"7");
      outtextxy(242,226,"8");
      outtextxy(235,197,"9");
      outtextxy(235,165,"10");
      outtextxy(260,140,"11");
      outtextxy(292,135,"12");
      gettime(&t);
      gotoxy(35,20);
      cout<<int(t.ti_hour)<<":"<<int(t.ti_min)<<":"<<int(t.ti_sec)<<"";
```

```
      i=(int(t.ti_sec)*(0.105))+11;
      j=(int(t.ti_min)*(0.105))+11;

      k=((int(t.ti_hour)*(0.105))*5+11);
       int min=int (t.ti_min);
       int rem=min/12;
      k=k+(rem*0.105);

      x=r*cos(i);
      y=r*sin(i);
      setcolor(14);
    line(300,200,x+300,y+200);

      xm=(r-5)*cos(j);
      ym=(r-5)*sin(j);
      setcolor(9);
    line(300,200,xm+300,ym+200);

      xh=(r-20)*cos(k);
      yh=(r-20)*sin(k);
      setcolor(6);
    line(300,200,xh+300,yh+200);
     delay(200.5);


    if(kbhit())
     exit(1);
      for( int pks=1;pks<=4;pks++)
     {
     l=200*cos(ii);
     m=200*sin(ii);

     setcolor(4);
     line(300,290,300+l,190+m);
     setcolor(14);
     setfillstyle(SOLID_FILL,14);
     circle(l+300,m+190,15);
     floodfill(l+300,m+190,14);

     ii-=0.105;
    delay(200.5);

    setcolor(0);
     setfillstyle(SOLID_FILL,0);
     circle(l+300,m+190,15);
     floodfill(l+300,m+190,0);
     line(300,290,300+l,190+m);
    }
   setcolor(0);
     line(300,200,x+300,y+200);
     line(300,200,xm+300,ym+200);
     line(300,200,xh+300,yh+200);
  }
 }}
```

CNC PROJECT:

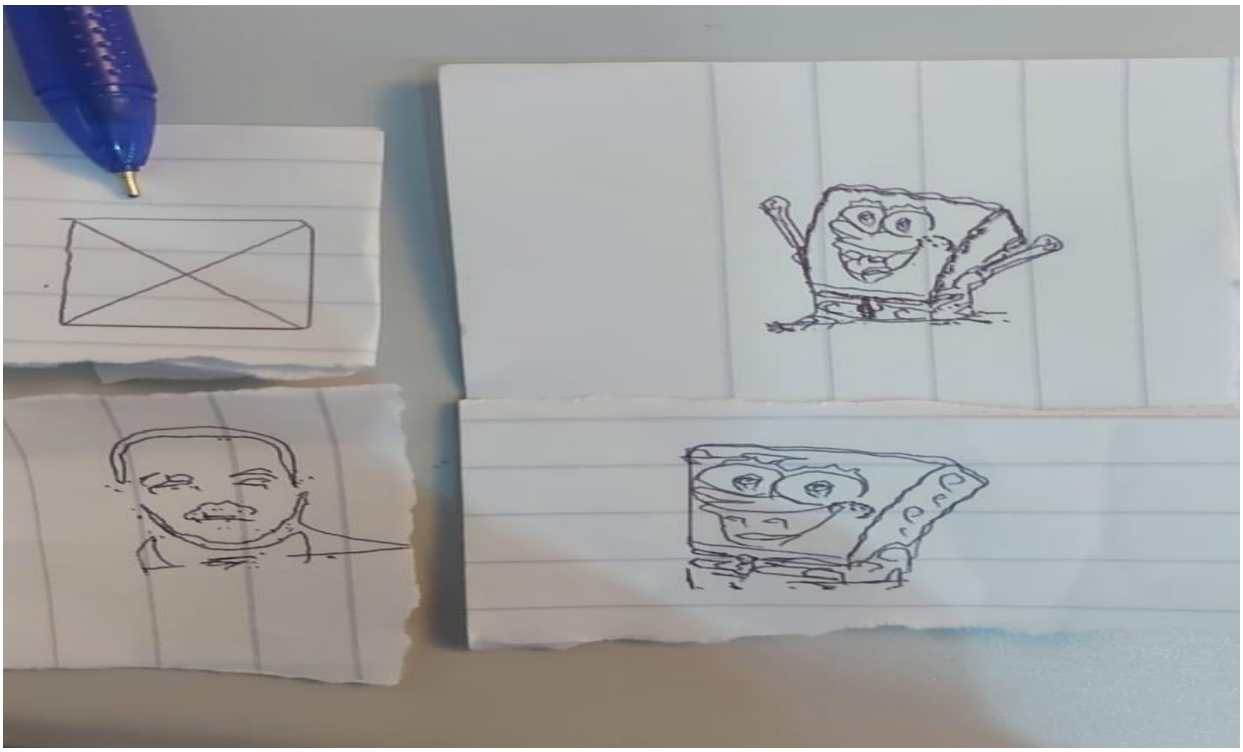


Circuit diagram:



List components:

Some figures from cnc:



CNC Code:

##########Arduino Code:

```
#include <Servo.h>
#include <AFMotor.h>

#define LINE_BUFFER_LENGTH 512

char STEfl = MICROSTEfl;

// Servo position for Up and Down
const  int penZUp = 110;
const  int penZDown = 50;

// Servo on flWM pin 10
const  int penServofl ni = 10;

// Should be right for DVD steppers, but is not too important here
const int stepsflerRevolution = 48;

// 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(stepsflerRevolution, 1);
AF_Stepper myStepperX(stepsflerRevolution, 2);
```

```
/* Structures, global variables */
struct point {
    float x;
    float y;
    float z;
};

// Current position of plothead
struct point actuatorflos;

// Drawing settings, should be OK
float StepInc = 1;
int StepDelay = 1;
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 StepsflerMillimeterX = 100.0;
float StepsflerMillimeterY = 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 = 40;
float Ymin = 0;
float Ymax = 40;
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 fl300 (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(penServoflin);
    penServo.write(penZUp);
    delay(100);
```

```
    // Decrease if necessary
    myStepperX.setSpeed(600);

    myStepperY.setSpeed(600);


    //  Set & move to initial default position
    // TBD

    //  Notifications!!!
    Serial.println("Mini CNC fllotter 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[64];                                 // Hope that 64 is enough for 1
parameter
    struct point newflos;

    newflos.x = 0.0;
    newflos.y = 0.0;

    //  Needs to interpret
    //  G1 for moving
    //  G4 fl300 (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(actuatorflos.x);
            Serial.print("  -  Y = ");
            Serial.println(actuatorflos.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 * StepsflerMillimeterX);
y1 = (int)(y1 * StepsflerMillimeterY);
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 o StepInc : -StepInc;
int sy = y0 < y1 o StepInc : -StepInc;

long i;
long over = 0;

if (dx > dy) {
    for (i = 0; i < dx; ++i) {
        myStepperX.onestep(sx, STEfl);
        over += dy;
        if (over >= dx) {
            over -= dx;
            myStepperY.onestep(sy, STEfl);
        }
        delay(StepDelay);
    }
}
else {
    for (i = 0; i < dy; ++i) {
        myStepperY.onestep(sy, STEfl);
        over += dx;
        if (over >= dy) {
            over -= dy;
            myStepperX.onestep(sx, STEfl);
        }
        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("flen up!");

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


    }
}
```

###############Java Code###############

```
import java.awt.event.KeyEvent;
import javax.swing.JOptionPane;
import processing.serial.*;

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 = "COM2"; // 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(null,
        "Select the serial port that corresponds to your Arduino board.",
        "Select serial port",
        JOptionPane.PLAIN_MESSAGE,
        null,
        Serial.list(),
        0);

    if (result != null) {
        portname = result;
        openSerialPort();
    }
}

void setup()
{
    size(500, 250);
    openSerialPort();
}

void draw()
{
    background(0);
    fill(255);
    int y = 24, dy = 12;
    text("INSTRUCTIONS", 12, y); y += dy;
    text("p: select serial port", 12, y); y += dy;
```

```
    text("arrow keys: jog in x-y plane", 12, y); y += dy;
    text("5 & 2: 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;
    if (key == '2') speed = 0.01;
    if (key == '3') speed = 0.1;

    if (!streaming) {
        if (keyCode == LEFT) port.write("G21/G90/G1 X-10  F3500\n");
        if (keyCode == RIGHT) port.write("G21/G90/G1 X10 F3500\n");
        if (keyCode == UP) port.write("G21/G90/G1 Y10 F3500\n");
        if (keyCode == DOWN) port.write("G21/G90/G1 Y-10 F3500\n");
        if (key == '5') port.write("M300 S50\n");
        if (key == '2') port.write("M300 S30\n");
        if (key == 'h') port.write("G90\nG20\nG00 X0.000 Y0.000 Z0.000\n");
        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("$16=0\n");
        if (key == '0') openSerialPort();
        if (key == 'p') selectSerialPort();
        if (key == '$') 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;
        stream();
    }
}
```

```
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: reallyo
}
```

Project video:

- https://drive.google.com/drive/folders/1Eiuu_gNcmJHGXiZHbSZL9IwFGUHyBSWj
- سيتم نشر فيديو أوضح للمشروع برسومات أخرى بشكل أفضل .