MICROPROCESSORS AND MICROCONTROLLER BASED DESIGN

For

B.E. Computer Engineering

Group Members			
Zakriya Asif Paracha 340413			
Aazain Umrani 350163			
Wajeeha Tahir 334584			
Degree	**42**	**Syndicate**	**B**

DEPARTMENT OF COMPUTER ENGINEERING
College of Electrical & Mechanical Engineering (CEME), NUST-Pakistan

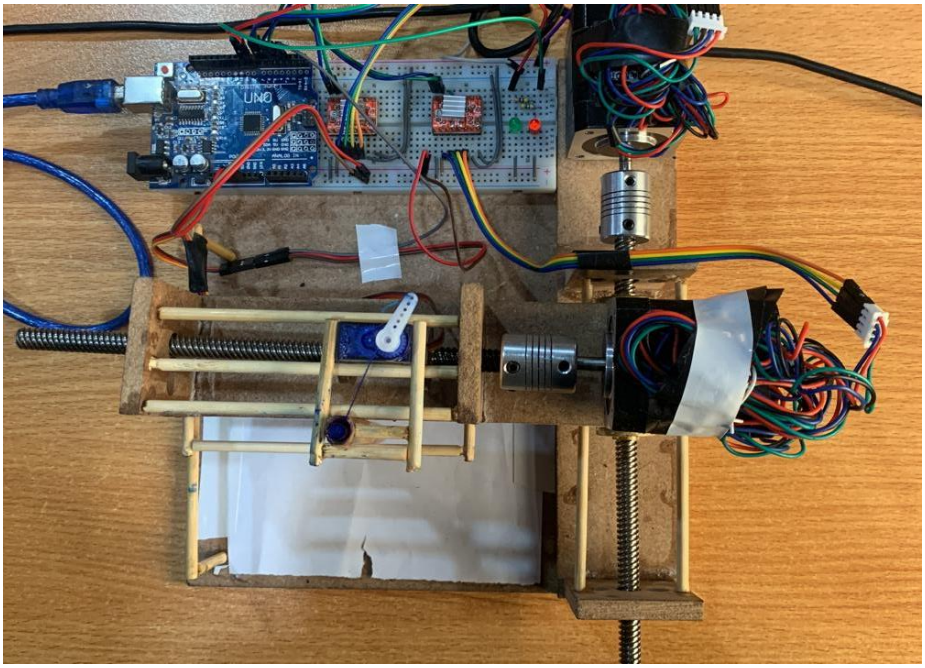
Final Report

2D Printer

Table of Contents

Introduction:

As our project we have made a 2D printer powered by Arduino UNO. The project is inspired by the 2D CNC machine.



There are two NEMA17 stepper motors, one for each dimension and a servo motor to plot the dots. The coding is not that complex, the actual complication lies in the structure and the hardware. According to the calculations, the motors accept around 1.2A (12V) while printing.

Internally, Uno has a 32x32 bool array. Which is printed /plotted one dot at a time. In the worst case, the system will take 2-4 minutes to draw the whole image. Then once the image has finished, the python code gets ready for the next image.

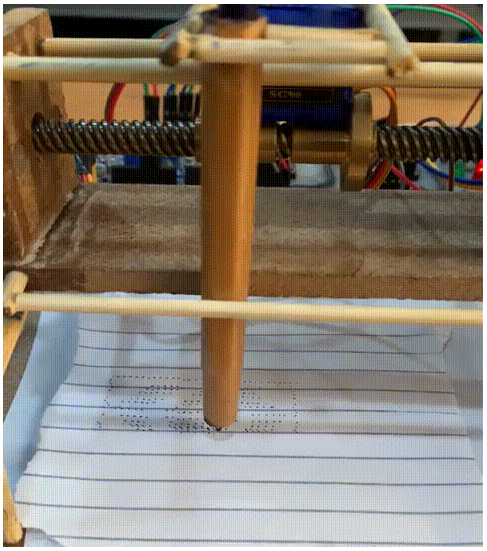
Working

Image Processing

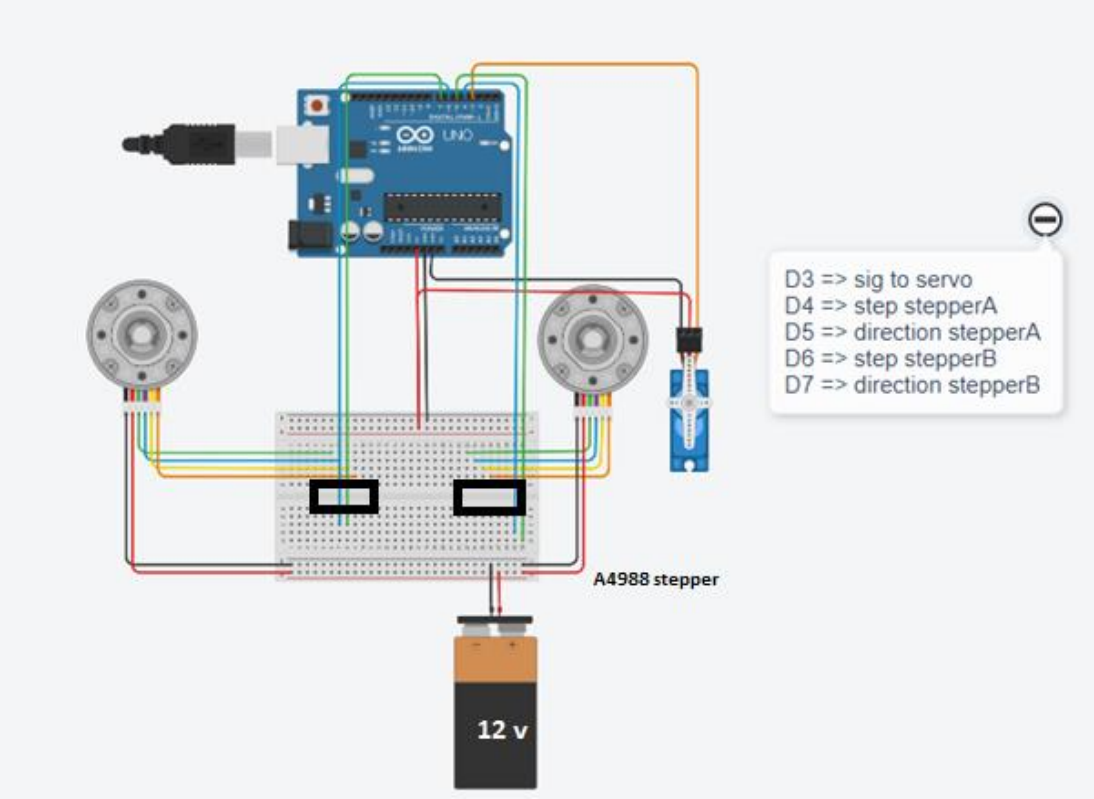
The Image module from the PIL library is used. The image is opened and converted to black and white. It is then resized to 32x32 pixels. The resulting image is plotted using the matplotlib library. The data is sent byte by byte to the Arduino serial port.

Printing

There is a platform that sits on the base. The platform has a stepper motor mounted on it that controls movement in the y-direction, a coupling connector connects the motor shaft to a threaded rod. There is a base mounted on the rod that moves with the rotation of the motor. A similar assembly is attached to the base of the platform that controls the displacement of the servo motor in the x-direction. The servo motor rotates to drop the pen on the paper and when it rotates in the opposite direction it lifts the pen off the paper. The stepper motors are controlled by a 12V power supply. The motor driver receives pulses from the Arduino. The servo motor is also controlled by the Arduino. The image is processed by a python program that takes a .bmp image and coverts it into a pixel array of 32x32 bits. The array is sent to the Arduino's serial port. When the Arduino is ready to read from the serial port a green led is turned on. The Arduino code reads the array from the serial port and converts it into a 2D array. The array is iterated in a nested loop. When a 1 is found the pen is dropped else it is lifted. During printing a red led is kept on. Once the printing is complete, the stepper motors return to the initial point.



Schematic



Code:

Image Processing:

```
#Import Libraries
from PIL import Image
import matplotlib.pyplot as plt
import numpy
import serial
import time

def printarr():
	print('{')
	for i in numpy.asarray(small_img):
		print('{',end='')
		k=0
		for j in i:
			k=k+1
			print(int(not(j)),end='')
			if(k<32):print(',',end='')
		print('},')
	print('};')

#Read image
img=Image.open('img.bmp').convert('1')


small_img=img.resize((32,32),Image.Resampling.BILINEAR)

#resize
o_size=(100,100) #output size
res=small_img.resize(o_size,Image.NEAREST)

#display image
print(">image will be similar to the plot! ")
plt.imshow(res)
plt.show()

#send to printer
print(">Initializing the print! ")
ser = serial.Serial('COM6', 9800, timeout=1)
time.sleep(0.02)

while (ser.readline() == b''):pass

for i in numpy.asarray(small_img):
	v   = 0
	p = 7
	for j in i:
		v=v + int(not(j))*pow(2,p)
		p=p-1
		if (p == -1):
			p = 7
```

```
                        ser.write(v.to_bytes())
                        time.sleep(0.01)
                        v=0



print("done sending data")
ser.close()
```

Arduino code:

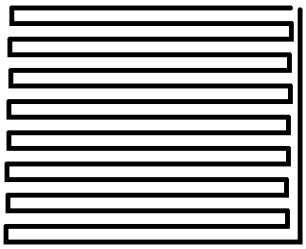
```
// the setup function runs once when you press reset or power the board
#include <Servo.h>
#define speed 2

#define pixeldensity 35 //1pixel at 20* distance
#define res 32

#define down 130
#define up 170
#define servodelay 150

bool pixelarray[res][res]={{0}};
Servo myservo;  // create servo object to control a servo

void setup() {
    Serial.begin(9600);

  // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
    //mot1- yaxis
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    //mot2- xaxis
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    //status led
    pinMode(8, OUTPUT); //red
    pinMode(9, OUTPUT); //green

    digitalWrite(9, HIGH);
    myservo.attach(3);  // attaches the servo on pin 3 to the servo
object
    myservo.write(up);

  }

void stepout(int mot,int steps)
{
  //led on
  int dir_pin,stp_pin;
  if(mot==1){dir_pin=7;stp_pin=6;}
  else if(mot==2){dir_pin=5;stp_pin=4;}
```

```
  digitalWrite(dir_pin, HIGH);
  //steps
  for(int i = 0; i < steps*pixeldensity; i++){
  digitalWrite(stp_pin, HIGH);
  delay(speed);
  digitalWrite(stp_pin, LOW);
  delay(speed);
  }
  }


void stepin(int mot,int steps)
{
  int dir_pin,stp_pin;
  if(mot==1){dir_pin=7;stp_pin=6;}
  else if(mot==2){dir_pin=5;stp_pin=4;}

  digitalWrite(dir_pin, LOW);
  //steps
  for(int i = 0; i < steps*pixeldensity; i++){
  digitalWrite(stp_pin, HIGH);
  delay(speed);
  digitalWrite(stp_pin, LOW);
  delay(speed);
  }
}

void dot()
{
          myservo.write(down);
          delay(servodelay);
          myservo.write(up);
          delay(servodelay);
}

//debug code
void printarr()
{
  for (int i=0;i<32;i++){
    for (int j=0;j<32;j++){
        Serial.print(pixelarray[i][j]);
        Serial.print(" ");
  }
    Serial.println();
    }
}


// the loop function runs over and over again forever
void loop() {
  int i= 0;
  int j= 0;
  //waiting for the serial data to be loaded
```

```
  Serial.println("Enter the data!");

  while(i<32){
      while (Serial.available()) {
        // get the new byte:
      char c= (char)Serial.read();
      for (int k=0;k<8;k++)
        {
        pixelarray[i][j] = ((c << k)  & 0x80);
        j++;
        if(j==32){j=0;i++;}
        }
      if(i==32){Serial.flush(); }

      }
  }


  //red led for printing
  digitalWrite(9, LOW);
  digitalWrite(8,HIGH);

  //acutual printing algorithm
    for (int i= 0; i < res; i++)
    {
      for (int j = 0; j <= res-1; j++){
          if (pixelarray[i][res-1-j]){dot();}
          if(j!=31)stepout(2,1);
          }

      stepout(1,1);
      i++;

      for (int k= res-1; k >= 0; k--){
          if (pixelarray[i][res-1-k]){dot();}
          if(k!=0)stepin(2,1);
          }

      stepout(1,1);
     }
    stepin(1,31);
    //red led for printing
    Serial.println("Done Printing");
    digitalWrite(9, HIGH);
    digitalWrite(8,LOW);
}
```

Algorithm:

We have chosen the scanning algorithm which follows a snake pattern and plots all the points. This gives the maximum time efficiency without optimization.



Specifications:

Images with dimensions of 8x8cm have been printed, within 2-4 minutes. The pixel density is 35 steps per pixel which is approximately 1mm.

```
#define speed 2 //delay between the movement between each pixel

#define pixeldensity 35 //1pixel at 20* distance
#define res 32 // number of total pixels

#define down 130 // servo degree at which the pen touches the page
#define up 170// degree at which pen is lifted
#define servodelay 150
```

Structure:

The structure is made from wood. A 22x20 cm board is used as a base. The 20x4.5 and 22x5 cm pieces are used as bases for the motor assemblies. The 4.5x5 and 4.5x4.5 cm pieces are used as walls. Holes are made into the walls to support the threaded rods. Another 4.5x5 cm piece is used as a platform for the second motor. The platform is supported using wooden skewers glued to the walls.

Hardware:

Electronics:

- Arduino Uno
- 2x Nema 17 Stepper Motors (12-24V)
- 2x A4988 Stepper Motor Driver (3-5V and 12V)
- Servo Motor (3-5V)
- Jumper Wires
- 12V Power Supply
- Breadboard
- 2x LEDs

Mechanical parts:

- Wooden Skewers
- Threaded Rods
- Nuts

- Wood Cutouts
    - 1x (22x5) cm
    - 3x (4.5x5) cm
    - 1x (4.5x20) cm
    - 2x (4.5x4.5) cm
    - 1x (20x22) cm
- Pen assembly

Conclusions:

It was a fun project, which gave us a lot if insight into mechanics, mathematics and motors. We have learned to understand and work with the serial communication of the Arduino. It involved some complex conversions from data types as we needed to cast the datapoints from bools to bytes and convert it back from bytes to bool and store it in the array in the Arduino.

Difficulties:

We faced three types of difficulties with this project.

1. **Physical stability.**

   As we designed and fabricated the hardware ourselves, maintaining the structural integrity and stability was a difficult task as a shaky structure would not produce a clear image.

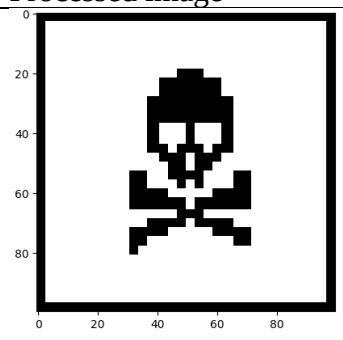
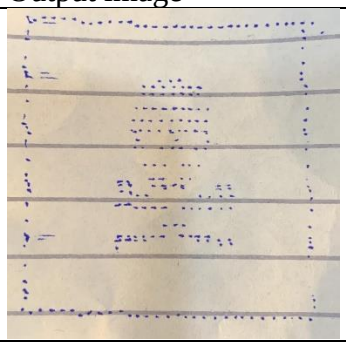
2. **Hardware datasheets.**

   The stepper driver (A4988) is sensitive, so we needed to go through its data sheets to learn about its actual specifications. Pico's 3.3V logic did not provide sufficient power to the drive hence we had to use an Arduino UNO instead.

3. **Serial communication**

   Interfacing Arduino with python was a key point of our project as it gave a great protocol to work, and we can extend the project in many directions. It took a lot of time devising a protocol which transfer the data from python to Arduino.

Outputs:

Input image	Processed image	Output image
		

References:

- Datasheet to A4988 driver
- Tutorial to stepper motor
- Example1 of our idea
- Example2 of our idea