



Republic of Iraq



Ministry of Higher Education and Scientific Research

University of AL-Qadisiyah

College of Computer Science and Information Technology

Multimedia department

Design and implementation Makelangelo Plotter

A graduation project is submitted to the Multimedia
department in partial fulfillment of the requirements for the
degree of Bachelor in computer science

By.

Mustafa Abdulkareem

Mjtaba Zuhair

Mustafa Qais

SUPERVISOR

Assist Prof Ali Muhsin Al-Juboori

2018

2017

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

قَالُوا سُبْحَانَكَ اللَّهُمَّ إِنَّا نَعْلَمُ أَنَّكَ مَا عَلَّمْنَا إِنَّكَ أَنْتَ

الْعَلِيمُ الْعَظِيمُ

صدق الله العلي العظيم

سورة النقرة: الآية (٣٢)

Contents

1.Introduction	2
2. Theory	4
2.1. Vertical Plotter.....	4
2.1.1. Arduino.....	5
2.1.2. Makelangelo	13
3. Design and implementation.....	16
3.1.Computer part.....	18
3.1.1.User interface	18
3.1.2. Programming Arduino	19
3.2.Microcontroller part.....	20
3.2.1.Software	20
3.2.2.Hardware.....	20
4.Test and result	27
5. Conclusion and future work.....	28
5.1. Conclusion.....	29
5.2. future work	30
References:	31

Figure 1 Componentes of V plotter	16
Figure 2 User interface of Makelangelo	18
Figure 3 User interface of Programming Arduino.....	19
Figure 4 9V AC-DC Adapter.....	20
Figure 5 Arduino Uno R3	21
Figure 6. stepper motors (B1544521).....	22
Figure 7 Motor drivers L298.....	23
Figure 8 Pendulum	23
Figure 9 SG90 9 g Micro Servo	24
Figure 10 Timing belt.....	25

Chapter one

1.Introduction

Vertical plotters, although being a slow and raster based device, are highly versatile drawing machines capable of extending their use in wide variety of fields. An artistic recreation, rather than an exact duplication, of digital images on canvas using elementary hardware components and complex algorithms act paradoxical to the way normal printers work. Yet their immense scalability and cost-effectiveness provide advantage over printers in certain areas [1] .

The vertical plotter is a kind of printer that is very much appreciated by professionals, since it does not print but draws by means of pens, and exactly in the same way a person would do with his hand; it allows to draw diagrams and sketches of various kind and with the maximum resolution, since it composes texts and images with a continuous line (and not with a dot matrix, as the impact, laser and inkjet printers do) [2].

And because of the ease of use and learning of Arduino by professionals and beginners and the abundance of sources and explanations available to him in addition to the possibility of use on several platforms and cheap price, which are two important characteristics distinguish from the peers of the controls and thus is the best choice for use with the vertical plotter ,Usually the price of it about \$ 300-400, but using the Arduino reduced this cost to about \$ 50,The program uses the Makelangelo to send the drawings to Arduino, which offers many options for drawing methods adds flexibility to choose the best result [3].

Chapter two

2. Theory

2.1. Vertical Plotter

Vertical Plotter is a small device able to generate a variety of graphics outputs, transforming every wall into a low refresh rate screen. In fact, it isn't but a small device, a "wall rover" travelling on a vertical surface. You write a note or take a picture then, through an computer, just send it to the plotter, which immediately turns the digital data into reality. The aim of its creators was to produce a device able to transform any surface into a huge monitor, allowing people to create art works, shopping lists or memoranda on every surface of their house or workplace. Simplicity and efficiency were the core guidelines. Starting from the experience gained on previous models, they designed a plotter able to cover large surfaces using a minimal amount of components and offering the maximum variety of different graphical outputs. Based on a lasercut chassis made from MDF or plexiglass, the design of the Vertical Plotter has sought to minimize the number of components needed and simplify as much as possible its assembly and use[4].

The main innovations which distinguish this brand new model from the older ones are: two on-board motors which stabilize the plotter and make its installation fast and simple, requiring no external components; the use of Processing for the generation of content and framework and of the Arduino platform for the hardware side, which make the plotter easily reproducible, user-friendly and less expensive; and the erasing system, allowing us to transform potentially every surface in a low-rate refreshable digital screen. The Vertical Plotter is a mediator between the digital and the physical dimensions. With it digital technology presents itself as a contents catalyzer, enhancing our life and making our everyday objects and environment smarter, more interactive, more and more sensitive. Vertical Plotter is an open-source project which during a few years of life featured the development of different prototypes of vertical plotters. [4]

This machine is divided into several parts:-

1- The part specialized in strict microcontroller Arduino.

2- The part concerned with processing the image and then sending it to the Arduino using Makelangelo program [4].

2.1.1. Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike[5].

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide [5] .

Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost

scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community [5] .

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- ❖ Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50.

- ❖ Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- ❖ Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

❖ Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

❖ Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money [5].

Arduino board is an open-source platform used to make electronics projects. It consists of both a microcontroller and a part of the software or Integrated Development Environment (IDE) that runs on your PC, used to write & upload computer code to the physical board. The platform of an Arduino has become very famous with designers or students just starting out with electronics, and for an excellent cause [3].

Different Types Of Arduino Boards

- Arduino Uno (R3)
- LilyPad Arduino
- Red Board
- Arduino Mega (R3)
- Arduino Leonardo

Features of Arduino Boards

Arduino Board	Processor	Memory	Digital I/O	Analogue I/O
Arduino Uno	16Mhz ATmega328	2KB SRAM, 32KB flash	14	6 input, 0 output
Arduino Due	84MHz AT91SAM3X8E	96KB SRAM, 512KB flash	54	12 input, 2 output
Arduino Mega	16MHz ATmega2560	8KB SRAM, 256KB flash	54	16 input, 0 output
Arduino Leonardo	16MHz ATmega32u4	2.5KB SRAM, 32KB flash	20	12 input, 0 output

We use Arduino Uno (R3)

The Uno is a huge option for your initial Arduino. It consists of 14-digital I/O pins, where 6-pins can be used as PWM(pulse width modulation outputs), 6-analog inputs, a reset button, a power jack, a USB connection and more. It includes everything required to hold up the microcontroller; simply attach it to a PC with the help of a USB cable and give the supply to get started with a AC-to-DC adapter or battery [3].

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, 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.. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start

over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

The Arduino Uno can be programmed with the (Arduino Software (IDE)). Select "Arduino/Genuino 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 preprogrammed 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 using Arduino ISP or similar; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. 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 reseing 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 [6].

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. Warnings The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed. Differences with other boards 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. Power The Arduino Uno board 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 center-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 from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become 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:

1. Vin. The input voltage to the Arduino/Genuino 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.
2. 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
3. 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
4. GND. Ground pins.
5. IOREF. This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

Memory The ATmega328 has 32 KB (with 0.5 KB occupied by 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 See the mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical. PIN MAPPING ATmega328P Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller [6].

In addition, some pins have specialized functions:

1. 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.
2. 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 `attachInterrupt()` function for details.
3. PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` function.
4. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
5. LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
6. TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled 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 it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with `analogReference()`.

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 Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, 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 (IDE) includes a serial monitor which allows simple textual data to be sent to and from the 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 SoftwareSerial library allows serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a

computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details [6].

2.1.2. Makelangelo

- ✚ The Makelangelo is an open source art robot that draws posters and murals. It's a fun, easy, and affordable way to learn about 3D printing CNC technology without the burning hot danger nozzle or the expensive, tricky plastic.
- ✚ The Makelangelo is fun. It can be used by children as young as six with adult supervision. Choosing a drawing and a color is easy – it will work with most Sharpie markers, Crayola markers, and whiteboard markers.
- ✚ The Makelangelo is risk-free. Print failure on a 3D printer can be expensive and dangerous. Print failure on a Makelangelo costs a piece of paper.
- ✚ The Makelangelo is affordable. A full 3D printer uses expensive plastic that won't fit in a school budget. The Makelangelo only uses markers and paper. It can even be mounted on a whiteboard so it takes up no room in a class.
- ✚ The Makelangelo is educational. It can be hard to make STEAM – Science, Tech, Engineering, Art, and Math – relevant to young minds. A Makelangelo won't run without trigonometry, pi, or basic algebra. It's easy to use the robot to make STEAM lessons relevant and timely. We've made sure that every lesson learned running a Makelangelo will give a student the experience they need to tackle harder subjects later.
- ✚ The Makelangelo software is the heart of the machine. It does all the heavy lifting.

The very first Makelangelo robot was “hard coded” to run a few simple lines. Then it was taught to understand gcode, the language of manufacturing CNC robots like 3D printers. That was a big improvement, but someone still had to sit there and type all the gcode into the robot, one line at a time. The company built the first Makelangelo app to deliver thousands of lines of gcode for us, as fast as the robot could draw. ...but turning a picture into gcode instructions took three apps and 40 minutes.

So The company streamlined the process into one app and now it's a few simple clicks to transform your JPEG, PNG,BMP, GIF, or DXF into gcode and start drawing [7].

Makelangelo program allows you to choose the height and width of the machine in addition to the dimensions of the paper and you can control the speed of drawing .

After selecting the image to be drawn, the program converts it to a format that the program can draw it which calls the Polar graph and there are several styles of conversion:-

- ❖ Boxyy
- ❖ Multipass
- ❖ Crosshatch
- ❖ Pule line
- ❖ Sandy Noble
- ❖ Spiral
- ❖ Wander
- ❖ Pulsing Spiral
- ❖ Voronoi stipples
- ❖ Voronoi zigzag

In addition we can write a text and send it to the Arduino to draw it [8].

Chapter Three

3. Design and implementation

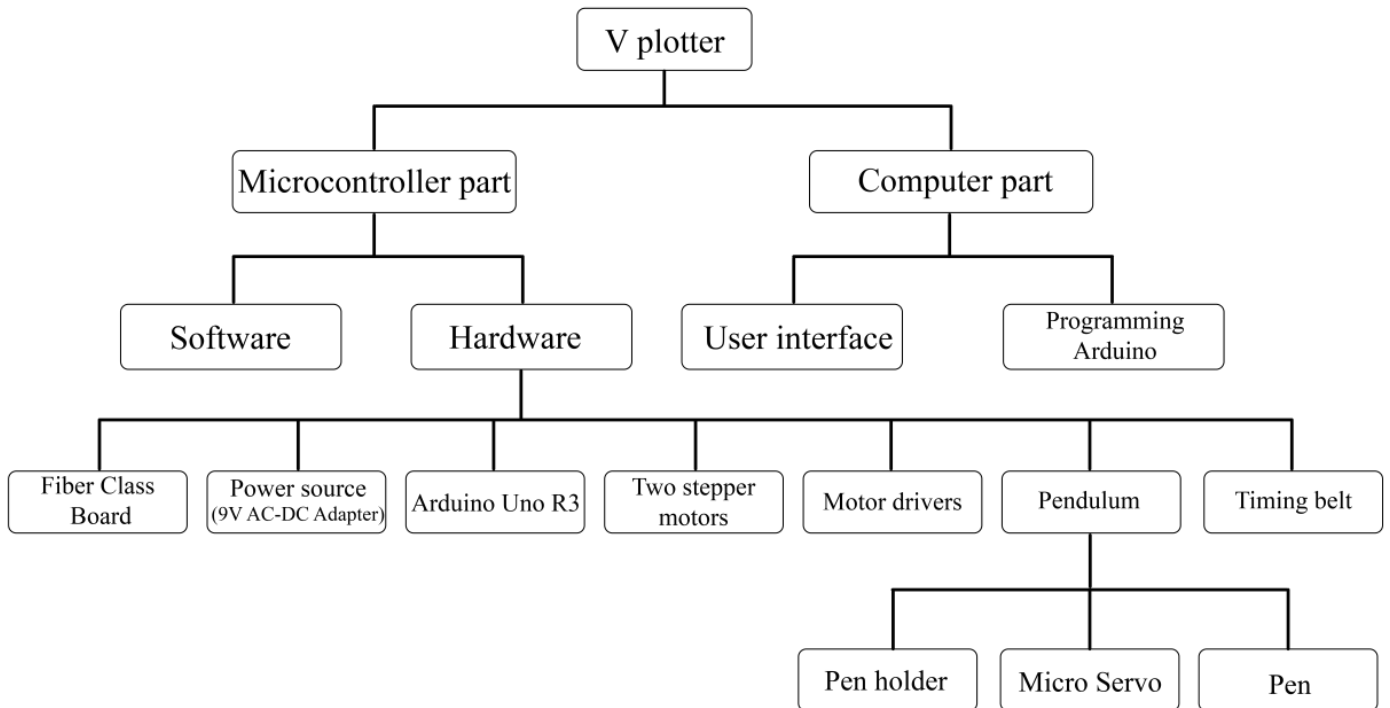


Figure 1 Componentes of V plotter

The algorithm

1. Receiving the paper size from the Makelangelo interface.
2. Receiving drawing settings:-
 - a- The value of the rise and fall pen.
 - b- The value of delay pen.
 - c- The X and Y coordinates of the starting point.
3. Receiving speed of stepper motors.
4. Receiving the coordinates of one step at a time.
5. The stepper motors start moving the pen to the position of the selected step.
6. The micro servo start working and put the pen on the paper.
7. The stepper motors moves the pen while it touches the paper to draw the current step.
8. When the drawing of current step is finish the stepper motors stop working and the micro servo lifts the pen.
9. If this step is the last go to (11) .
10. If not go to (4) to drawing the next step and go on.
11. The stepper motors moves the pen back to its original point.

3.1.Computer part

3.1.1.User interface

Using the Makelangelo program, which provides an interface to adjust the drawing settings in addition to converting the image into vector drawing according to many styles and then converting it to Gcode and sending it to Microcontroller (Arduino) to drawing it.

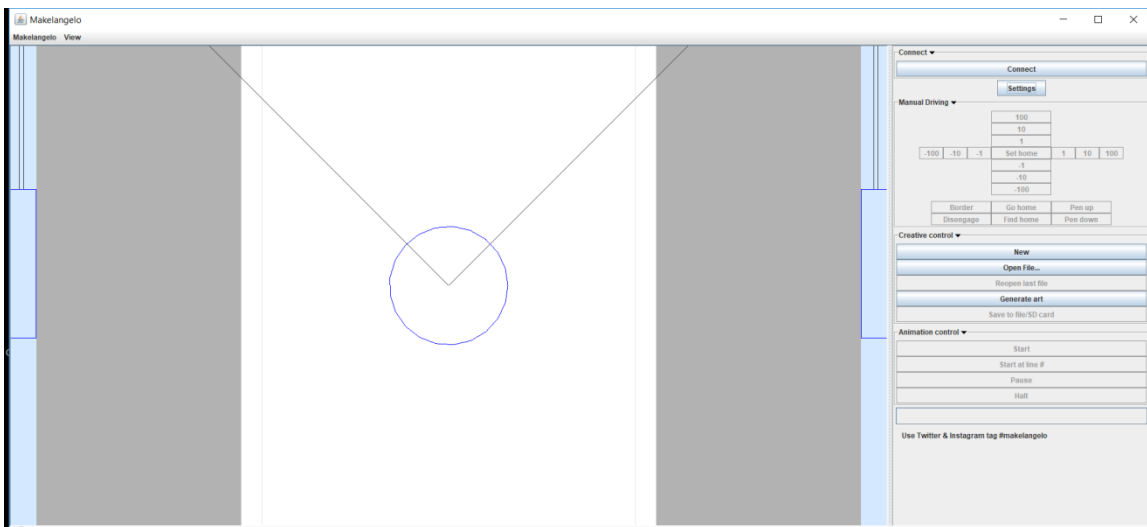


Figure 2 User interface of Makelangelo

3.1.2. Programming Arduino

Write the code that is placed inside the Arduino, which controls all the tools related to it according to the data that receive from the interface program to send the appropriate commands for drawing.

The code is written by the Arduino IDE editor.



Figure 3 User interface of Programming Arduino

3.2. Microcontroller part.

3.2.1. Software

The code is received from the computer and stored within the Arduino to be called at any time without the need to reload it from the computer.

3.2.2. Hardware

3.2.2.1. Fiber Class Board

The two stepper motors are mounted on the fiberglass board as well as set the paper that draws on it.

3.2.2.2. Power source (9V AC-DC Adapter)



Figure 4 9V AC-DC Adapter

3.2.2.3. Arduino Uno R3

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

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 by	32 KB (ATmega328) of which 0.5 KB used bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

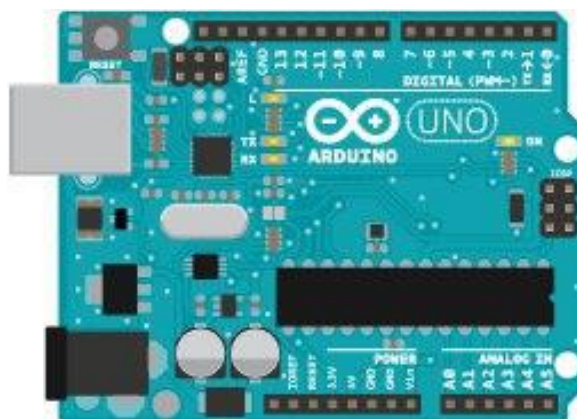


Figure 5 Arduino Uno R3

3.2.2.4. Two stepper motors

We will use two stepper motors (B1544521) .



Figure 6.a stepper motors (B1544521)

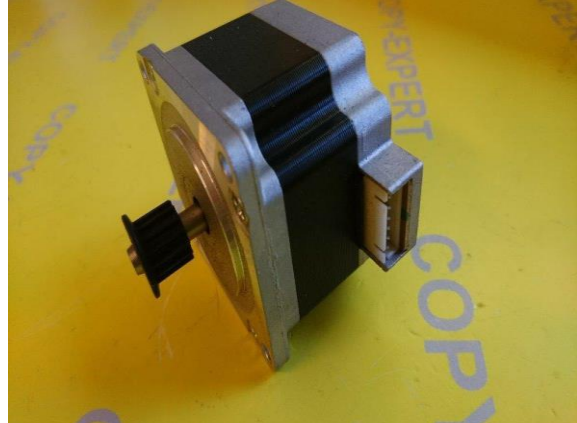


Figure 6.b stepper motors (B1544521)

3.2.2.5. Motor drivers

Matching the arduino with two stepper motors to drive it and here we use

DUAL FULL-BRIDGE DRIVER

The L298 is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

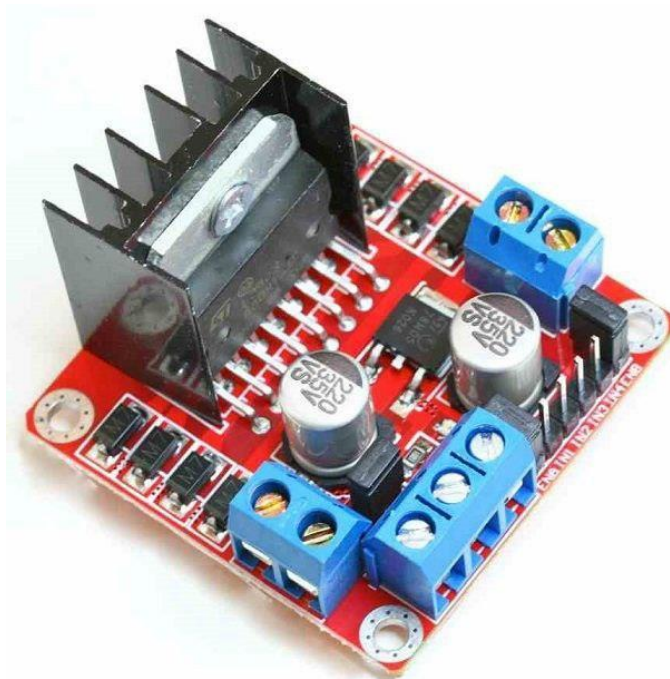


Figure 7 Motor drivers L298

3.2.2.6. Pendulum

Connect with motors to control of of it in x-axis and y-axis



Figure 8 Pendulum

The Pendulum Consists of three parts:-

- ❖ pen holder.
- ❖ Micro Servo

Which controls the z-axis that represent the pressing of pen on paper.

We use **SG90 9 g Micro Servo** that is a Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

Specifications

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 μ s
- Temperature range: 0 °C – 55 °C

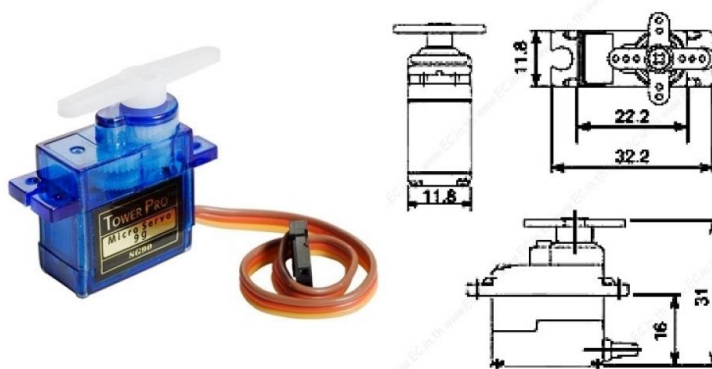


Figure 9 SG90 9 g Micro Servo

- ❖ Pen.

3.2.2.7. Timing belt

Which binds the pendulum to the motors to control the pendulum in the x-axis and the y-axis.



Figure 10 Timing belt

Chapter four

Test and Result

Chapter five

5. Conclusion and future work

5.1. Conclusion

- 1) When we used the Arduino microcontroller the cost was reduced from \$ 350-400 to about \$ 50.
- 2) Using vertical plotter, we can draw freely and easily without restricting space.
- 3) Easy availability of pieces and cheap prices.
- 4) The abundance of many types of physical parts that can be used instead of the parts currently used.

5.2. future work

- 1) The plotter can be developed in the future to paint in a color rather than a single color.
- 2) The pen can be replaced by a laser cutter to be an alternative to the cnc.
- 3) Develop the plotter to be moving instead of fixed on the fiber class can be drawn on any surface or wall.

References:

1. Low, D.A., *A Manual of Machine Drawing and Design*. 2009: Read Books Design.
2. SARCAR, M.M.M., K.M. RAO, and K.L. NARAYAN, *Computer Aided Design and Manufacturing*. 2008: PHI Learning.
3. Barrett, S.F., *Arduino Microcontroller Processing for Everyone!* 2010: Morgan & Claypool Publishers.
4. THYER, G.E., *Computer Numerical Control of Machine Tools*. 2014: Elsevier Science.
5. Seneviratne, P. *Building Arduino PLCs: The essential techniques you need to develop Arduino-based PLCs*. 2017.
6. Perea, F. *Arduino Essentials* 2015; Available from: <https://books.google.iq/books?id=NrjNBgAAQBAJ>.
7. Suh, S.H., et al. *Theory and Design of CNC Systems*. 2008; Available from: https://books.google.iq/books?id=c_-3TxZlnpMC.
8. Ablameyko, S. and T. Pridmore, *Machine Interpretation of Line Drawing Images: Technical Drawings, Maps and Diagrams*. 2012: Springer London.