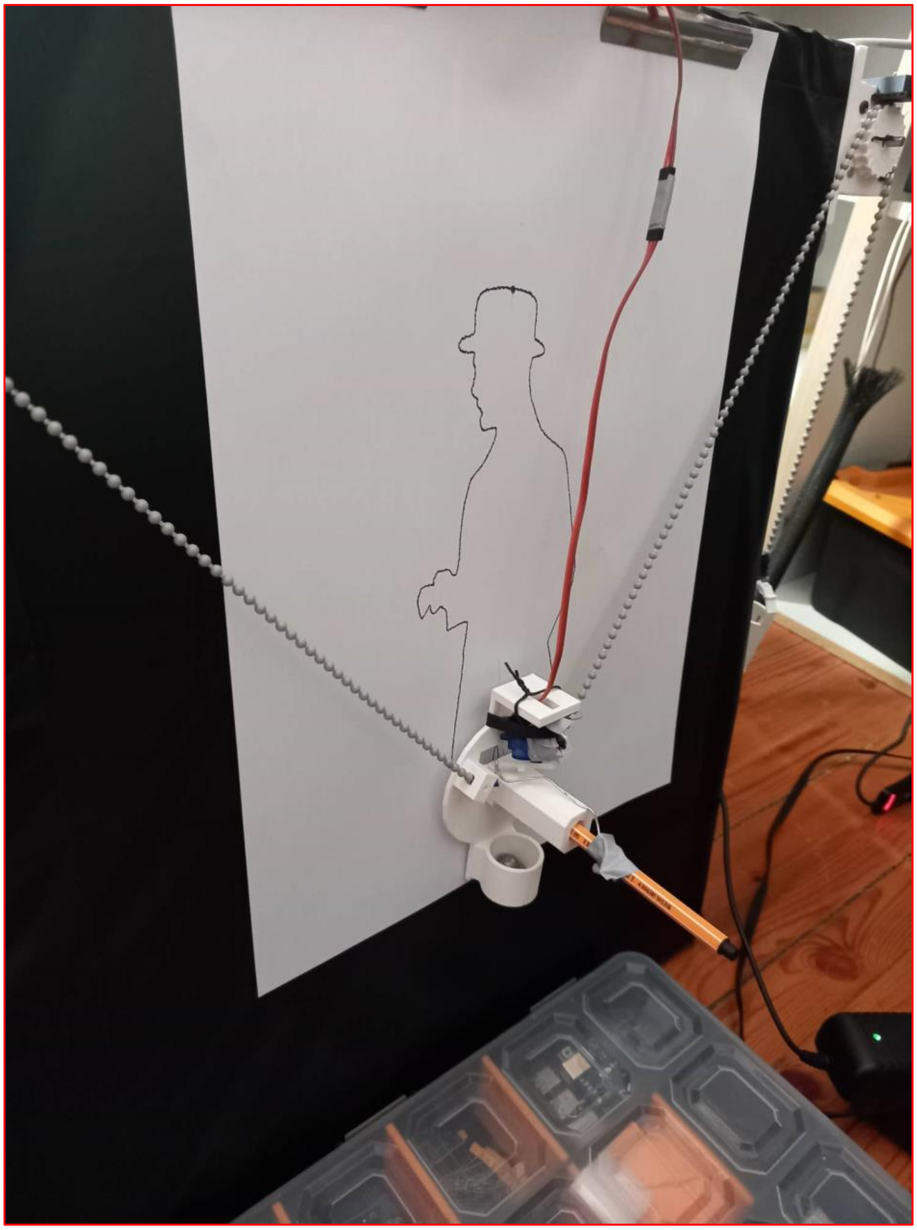
XY Plotter Drawing Robot Documentation



Implementation by George Zacharis

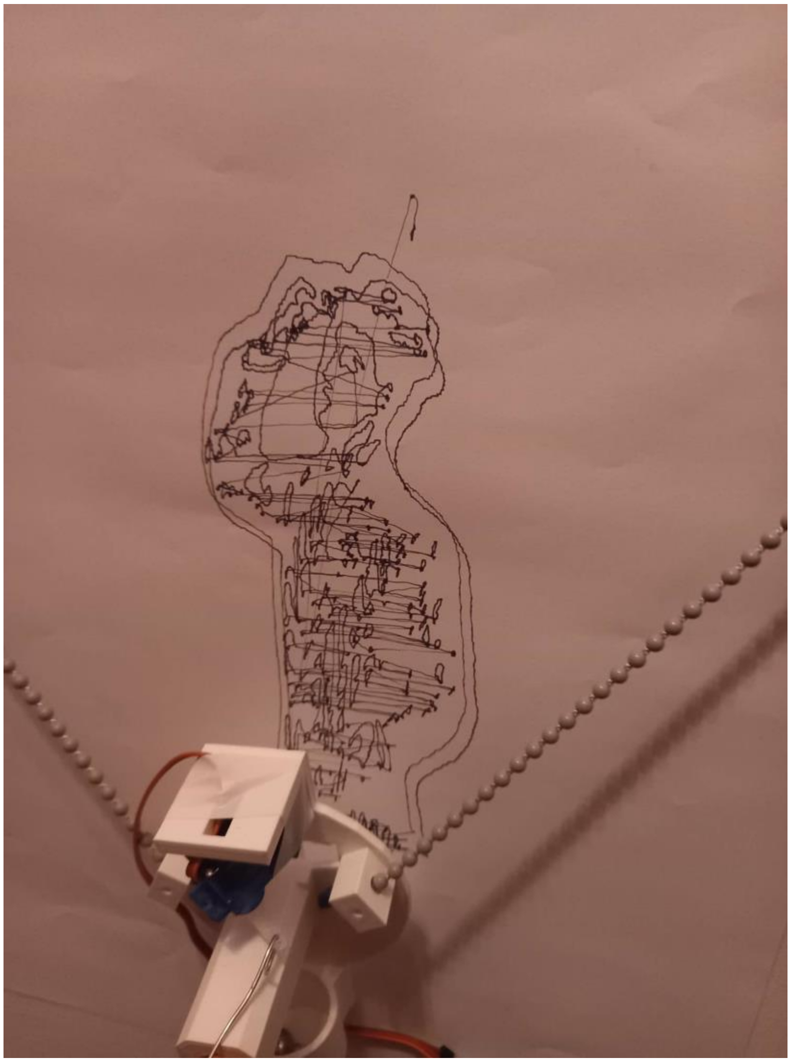
LinkedIn: https://www.linkedin.com/in/george-zacharis/

GitHub: https://github.com/TechZx

Index

Introduction

The present is a manual for the development of a XY Plotter drawing robot as well as a complete tutorial for setup, configuration, and troubleshooting. XY Plotter robots draw imported images using only 2 axes X and Y thus its name. It is an open-source project and there are many implementations of such robots as well as some community forums for questions. The purpose of this documentation is to comprise a complete and solid manual for XY Plotter robots, having collected information from various other sources as well as this present robot.



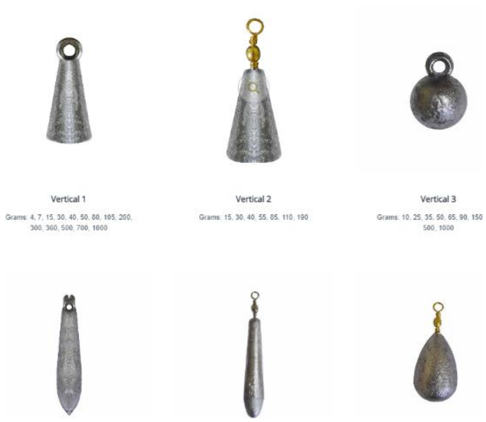
Hardware Requirements

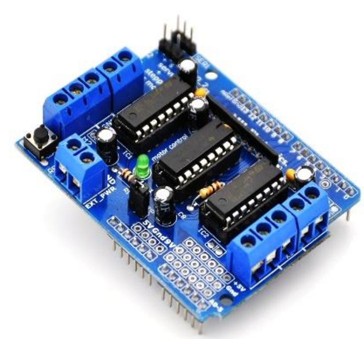
Bellow there is an index of the hardware required to develop a XY Plotter robot. Note that there are many variations of the hardware used in various robots, so this is not an exclusive list. This is the list of the hardware used to develop the robot of this manual as well as links to the products.

- Arduino Uno
- L293D Motor Driver Shield - Expansion Board For Arduino
- 2 x Nema 17 Stepper Motors
- Micro Servo Motor
- 12v 2A PSU
- 2m String Roll
- 2x Paper Grip Clips
- Black Drawing Canvas
- Fishing weights
- Some jumper cables
- Type B USB to USB (at least 2m)

Furthermore, for the needs of the project, some 3d printed parts were used. To be precise, the most 3d parts were used from the project of Met Arduino at Instructables.

Since there are many various products with common names here are some pictures of some of the ones mentioned above for reference:





Notes

- The fishing weights are used because they come in many sizes and forms and they are cheap, so this gives a freedom to experiment on what weight best suites each robot and its needs, so they are highly recommended.
- The stringed roll is used for convenience purposes. Instead of Stringed roll, a timing belt for motors can be used. Theoretically a timing belt with a proper aluminum timing pulley for the motors gives the robot a better accuracy, but practically for a simple project there is no big difference since the parameters can be tuned via the firmware which is explained later in the documentation.
- If 3d printing is convenient, the paper grip clips can be skipped and replaced with some 3d printed custom parts to hold the paper steady. This is recommended because this way the steadiness of the paper can be increased and also 3d printed custom parts may take less space in the construction which is for the best.
- The choice of a drawing canvas as a base is meant for the prototype robot and to explore the possibility of the electrical components to be fit on the back so that the robot looks like a wall frame construction. A wooden back may be a sturdier choice, but it may increase the weight and cost of the built. The choice and the proportions of the back depend on the manufacturer. The choice of the drawing canvas although works fine. A white canvas can easily get dirty either during built or tests or other causes, that’s why a black colored canvas is preferred.
- The 3D printed pulleys for the cords come with some disadvantages. They can be damaged and corrupt easily resulting in losing steps from the motors and not rolling as they are supposed to. Also they may need to be custom designed to the cord used so they have sufficient space for each step of the cord.

Assembly

The assembly part of the built is not that tricky but there are some things that need attention which will be explained. For a better understanding it is obvious that a visual aid will help a lot ,so it is recommended to also look in parallel this video by MERT Arduino & Tech which also was the base for this project. Make sure all the parts are gathered before assembly. Since the XY Plotter robots are named after the 2 axis and 2 motor principle applied, such robots can be built with different dimensions and hardware so unavoidably each built needs its own tuning and setup as for the assembly and parts except for the basic concept and assembly.

Motors

To begin with, the 3D printed motor parts need to be installed on the motors and screwed on the sides of the wood of the drawing canvas like the picture below. Make sure the motors are installed in the same high and facing the same way. It is also important, if you are using 3D printed pulleys, to make sure that they cannot move without the stepper motor moving. If they are loose, they need to be secured cause this will definitely cause problems to the functionality of the robot.

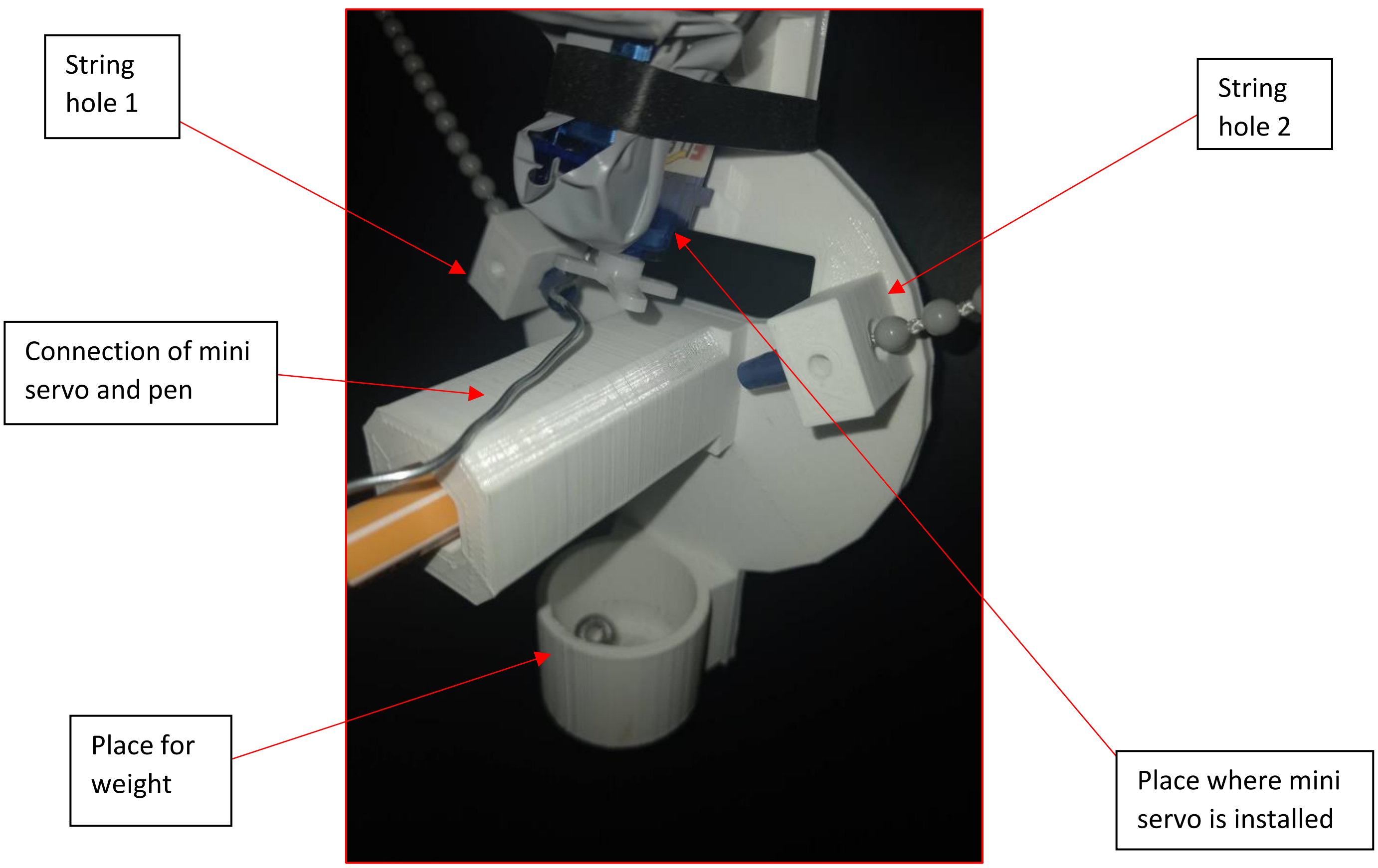
There can be noticed some black extensions on the top of the 3D motor holders. This is a way to prevent the string cords of going out of the way. There are many ways to do this, not necessarily with 3D printing, but it is highly suggested.

But in this step, they are not required to be installed yet.

Gondola

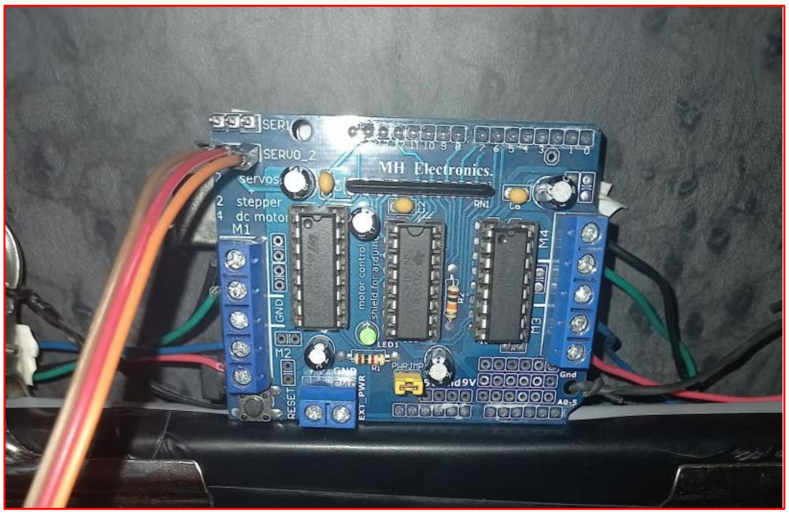
Next, the part that makes the drawing needs to be assembled, which is called the “gondola”. First the stringed cord or timing belt need to be cut in 2 even pieces. Calculate some extra cord from both sides of each piece for the next steps. An empiric way to measure the cord is to check how much the robot will need from each axis to reach to the farthest place of the robot, which is the bottom center. The cords in this measuring need to be stretched so it can be accurate. The analogy of the strings should be like the picture below. At the end of the cords install the 3D weight holders or some form of weights. The exact amount can be found in practice and depends on each robot.

The stringed cords need to be installed to the gondola from each side where the holes are. They need to be secured steadily. The mini servo needs to be installed securely and steadily in the place in between the string holes. This is a tricky part cause the servo needs to somehow to be attached with a pen so that the robot can lift and drop the pen when needed. As is shown in the pictures, the middle hole of the gondola is for the pen. It is suitable to install a thin pen that easily leaves ink for better results. As a quick and budget solution, a small paperclip can be adjusted to connect the servo with the pen. At last, the gondola part has a place for a weight so it can be steady and not leaving the surface of the canvas.


String hole 1
String hole 2
Connection of mini servo and pen
Place for weight
Place where mini servo is installed

Electronics & Motors

The motors we are using have 2 coils and each one outputs 2 wires. You can check the motors schematic to find the 2 pairs of wires. Another way is to use a multimeter. Once you do you need to connect them to the Arduino shield, which should be connected to the Arduino board. If the motors have any pins installed it is better if they are cut so there is complete access to the wires. It is also practical that the wires are burned with some solder and a soldering gun so they can be installed to the shield more easily. At first do not fully secure the electronics because there might occur some changes to the wires so the right way can be found so that they move the right way with the firmware. It might take a few tries. The mini servo is also connected to the shield. The servo will probably have short cables, so they need to be extended with some jumper cables, otherwise it will cause problems to the functionality of the robot. The gondola needs to be free to move without any constraints. At last, make sure you have access to the board so you can use the serial port of the Arduino and the power input.



Software Setup and Configuration

Software requirements

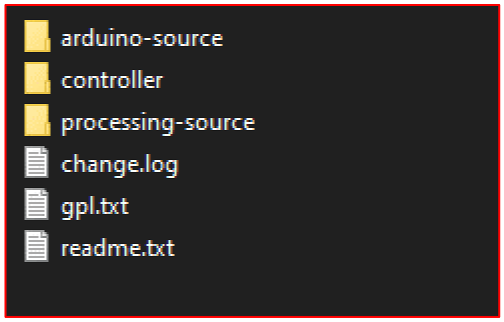
There is a community for such drawing robots for exchange of opinions, tips and questions. An open-source firmware has been developed for XY Plotter robots, called Polagraph. Also Arduino IDE and Processing needed. Before you download and install anything, some things need to be noted first.

Notes

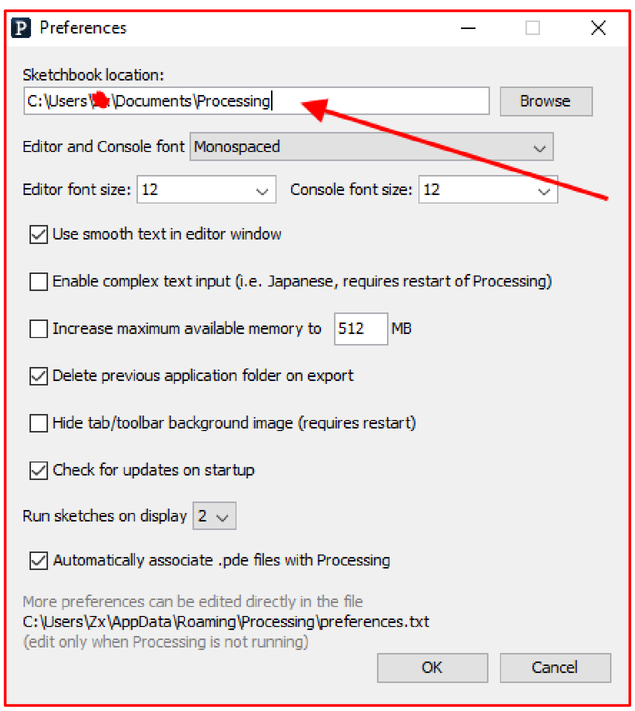
- A good practice for the project is to make a VM environment using Virtual Box for example to setup all the requirements and not have any effect in your system, but this is optional.
- It is suggested that everything concerning this project should be in the same folder for practical reasons.
- Make sure your PC has Java and JDK installed and updated.
- At the time this documentation is being written, many projects so far use Processing 1.5.1 version. But this version is not supported by Processing anymore and even if it can be found it has trouble when installed and compatibility issues with JDK. So, the next version is used which is **Processing 2.2.1** and it is almost an one-way road as the next versions, by the time of this documentation, have problems also with the Polygraph firmware. Lastly, it needs to be noted that the project was set up, thus the following steps, in **Windows**.
- For the Project, it is suggested that older versions of Arduino are more compatible with the project. In this specific project, there was found some truth in this suggestion concerning the libraries that need to be setup in the next steps. It is checked that Arduino versions 1.8.19 and 1.0.5 work. But Arduino version **1.0.5** is the safer choice.

Software setup

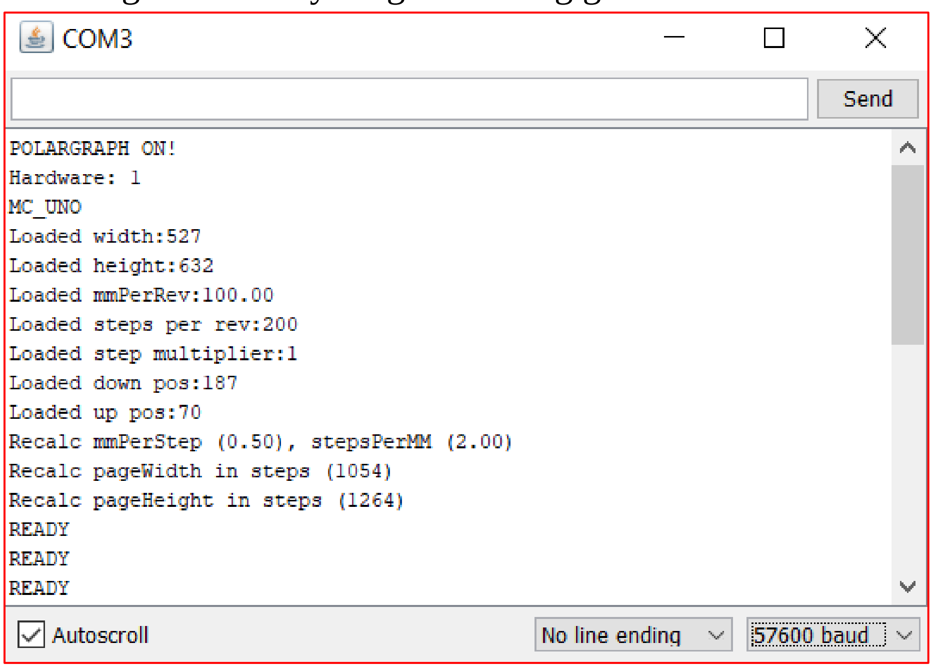
After the download of the software mentioned it is needed to first setup some things. The Polagraph zip file contains the following files:



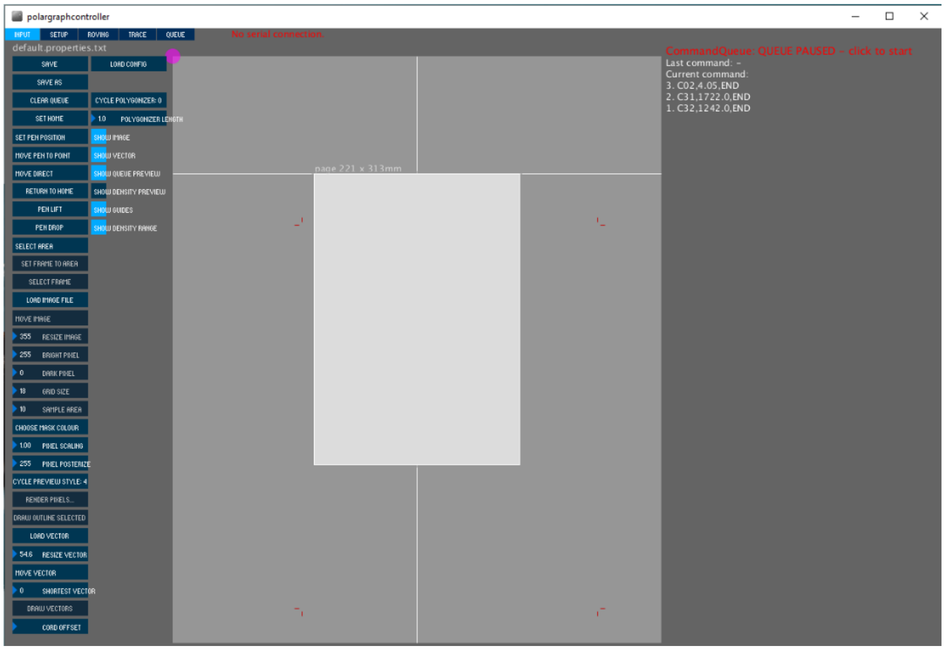
1. The libraries of the folder **arduino-source** located in the Polagraph zip need to be copied to the libraries file of the Arduino program which are usually found, if installed by default, in **C:\Program Files (x86)\Arduino\libraries**
2. Open processing and find the Sketchbook folder in **File-Preferences** and copy all the library files of the **Polagraph-processing-source-Processing libraries** folder in the Processing Sketchbook folder



3. Copy the file **Polagraph/processing-source/polagraphcontroller** in the Processing folder made by the program during installation. Usually it is located in the Windows **Documents** directory.
4. Upload the **polagraph_server_a1.ino** located at **\Polargraph 2017-11-01\arduino-source\polargraph_server_a1** directory to the Arduino of the robot. First choose your serial port at the settings of the Arduino IDE. It should be uploaded with no problem if everything is configured right. The Arduino program has some comments for direction, if needed, to setup some additional things like Arduino board model, motor shield model or motor pins. But it is highly suggested that first, if problems in upload occur, to check that all the steps mentioned before are completed right.
5. After upload, open serial and set the baud rate at 57600. There should be messages that everything is working good like this:



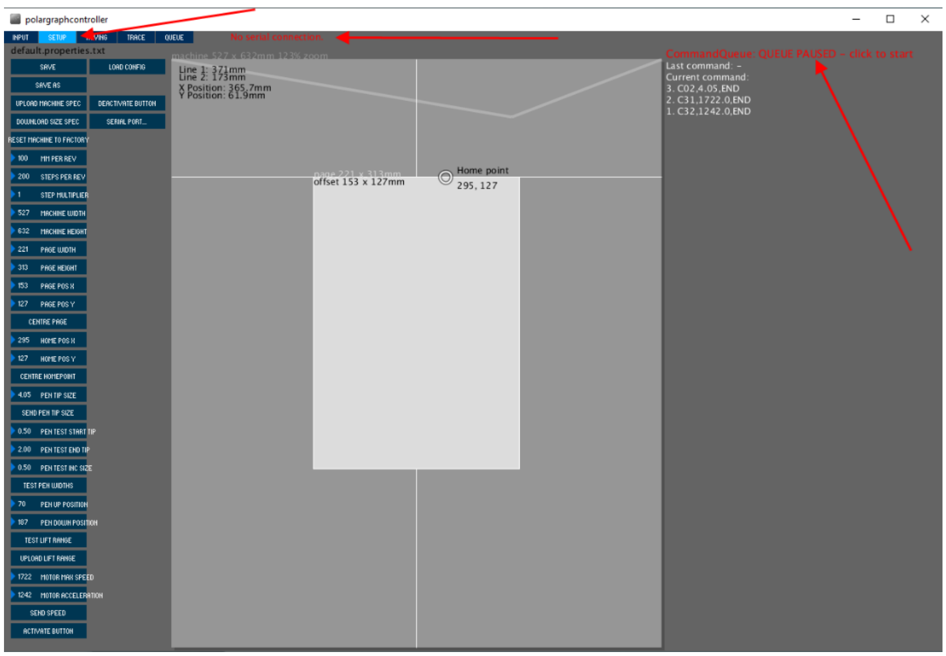
6. Now launch processing, go to **File-Sketchbook** and there should be the choice **polagraphcontroller.** Open the file and compile it and then the GUI of the polagraph should open and should look like this:



Polagraph GUI

Setup

In this section is explained the use of the GUI for a simple run and directions on how to configure the settings. On the right, the **Command Queue** section can be seen. To begin sending the command Queue to the robot, for the later steps, you need to click it to start. Firstly, you do not need to click it to start. To setup your machines values. This can be done in the **Setup**



1. Select in **Serial Port** the serial port your robot is plugged in.
2. Set your **STEPS PER REV** and **MM PER REV** according to your stepper motors specifications.
3. Measure your build's size and fix the values accordingly.
4. You will need to also set up a fixed home point where your gondola's position should have as a "Home position" in the firmware.
5. When the parameters are configured to your case you can save them and export them as a txt file so you can import them every time you want to use the robot. You can load your properties in **Load Config**.
6. When it is done you should click on **Upload Machine Spec**. You should see some commands appear in the Queue section.

7. Click the **Command Queue** to upload the machine specs. Then click it again to stop.

Input

In the **Input** section, the moves of the robot can be controlled. This part needs some different tuning in each robot.

1. To check how the gondola is moving you can click on **MOVE PEN TO POINT** and then click on the board so the robot follows the command.
2. Move the pen and try different positions until the gondola is in your "home point". You can do it in steps if there is no more space to move. Each time press **SET HOME** so there are checkpoints each time and calibrate it slowly.
3. Once the gondola is in the desired Home place both in software and hardware, click on **SET HOME**.
4. You can test the gondola's pen and drop by clicking on **PEN LIFT** and **PEN DROP**.
5. Select the area where the drawing will take place by clicking on **SELECT AREA**.
6. Then click on **SET FRAME TO AREA**.
7. Finally load an image for drawing , click draw **DRAW VECTORS** and click the **Command Queue** for the robot to start drawing**.**

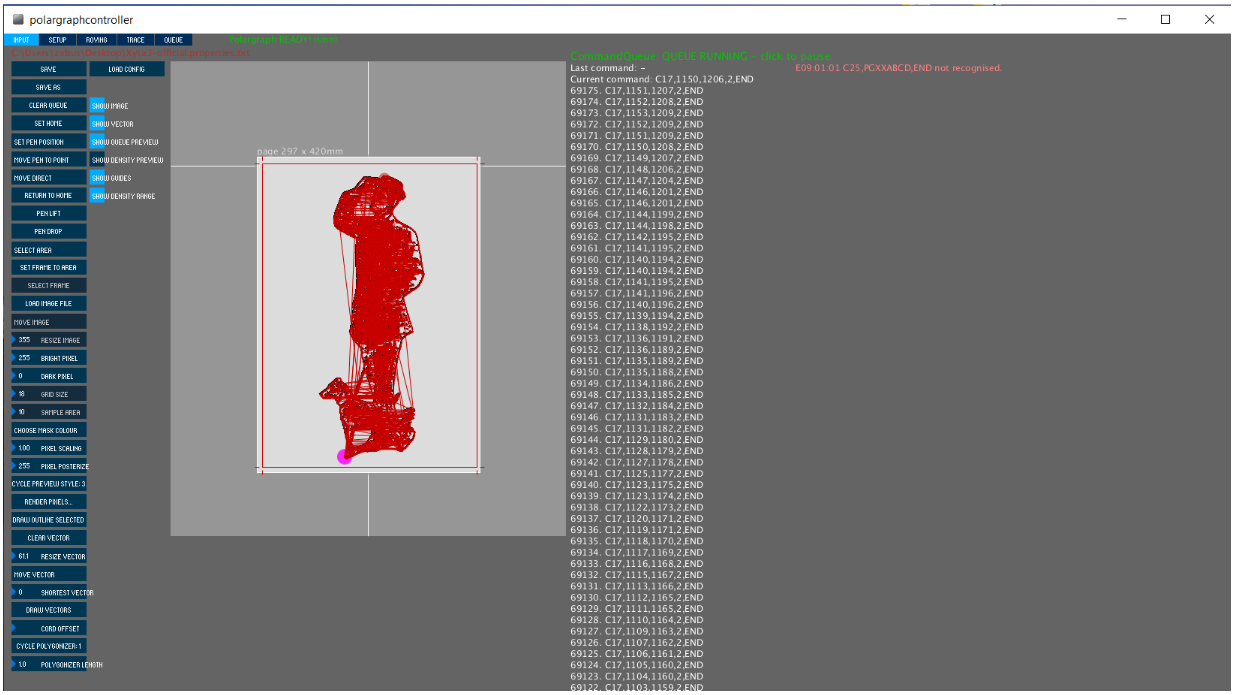
Notes

- It might take some time to get familiar with the GUI so you can control it more easily. For example, to change the parameters you need to move the mouse up and down.
- The GUI has some bugs (at least at the time of this writing) that will be mentioned here, so it needs some patience.
- In the **Setup** section, in **Serial Port…** there is a bug in choosing your Port. You need to initially select **NO SERIAL CONECTION** and then to select your port
- In the **Input** section, when loading a vector, it might take some time to load it or not load it. It might need to be clicked and loaded a second time.
- The **pink circle** on the map is showing where the software thinks your pen is located. This pink circle needs to be inside your red frame of your **selected area** of drawing or the software will assume that it is out of range and will not move your commands
- If the vector is loaded and it is not visible you can resize it and move it.
- The images need to be **.svg files**. There are many vector images on the internet to choose as well as many sites where svg images can be downloaded. In the zip file of the project there are some svg images included for demo.
- You can also convert a sketch image into .svg using a program like Photoshop.
- The drawing area and paper are up to the user's decision. A good recommendation is to use an A3 paper because it has a big surface for testing. You can easily find each paper size dimensions on the internet so you can import the values to the **Setup** section.

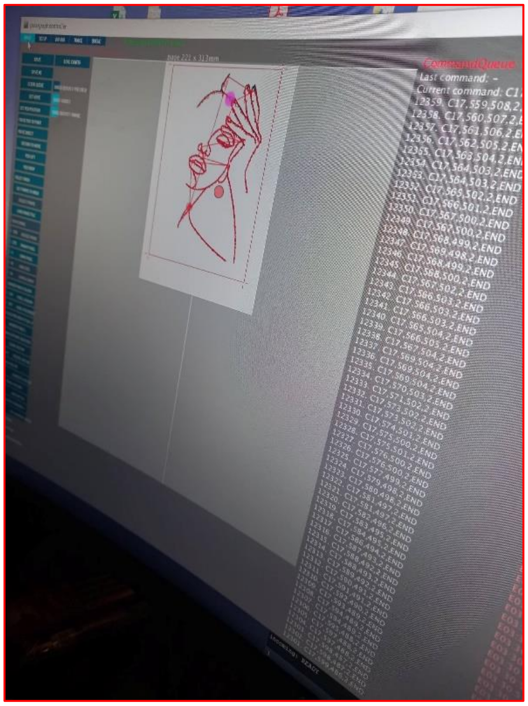
Recap for Run

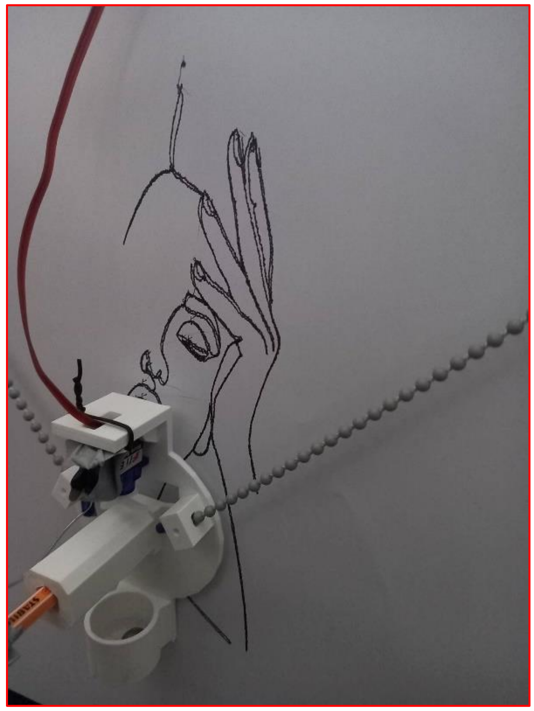
- Load config/upload parameters
- Upload machine specs
- Move pen to point/ Set Home
- Select area for drawing
- Set frame to area
- Load Vector
- Set vector to right position
- Draw vector

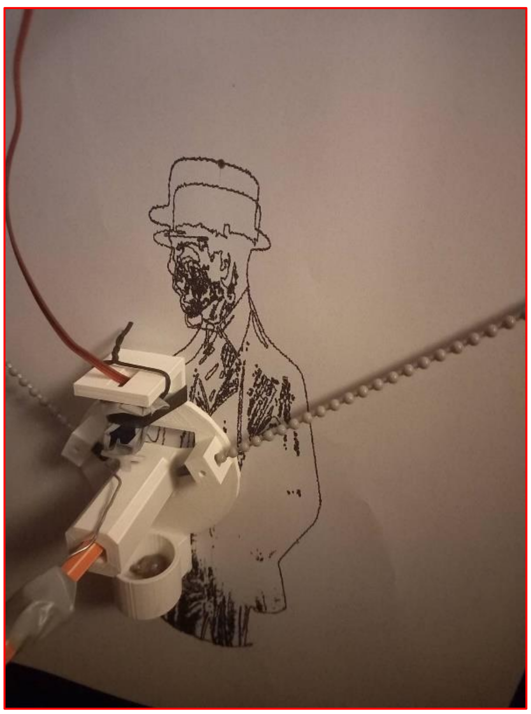
When running right the GUI should look like this:

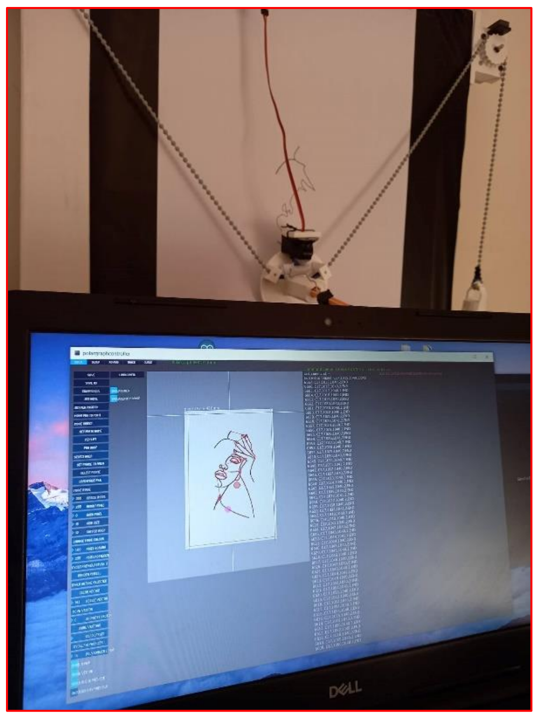


Images of Prototype









Specifications of prototype

List of hardware used

- L293D Shield : https://www.hellasdigital.gr/go-create/arduino-shields-and-accessories/motor-shield-board-module-l293d-for-arduino/
- L293D : https://www.hellasdigital.gr/electronics/motors-and-drivers/drivers/l293d-push-pull-four-channel-motor-driver-ic-st-dip-16/
- Arduino Uno : Arch. E.C.E. laboratory
- 2x Nema 17 stepper: https://grobotronics.com/stepper-motor-42byghw208-2.6kg.cm.html
- Micro stepper: https://www.google.com/url?sa=D&q=https://grobotronics.com/servo-micro-1.5kg.cm-plastic-gears-feetech-fs90.html&ust=1664985060000000&usg=AOvVaw0izsTb4rD4VEfcqZNBOA3G&hl=en&source=gmail
- 12v 2A PSU: https://grobotronics.com/power-supply-12vdc-2a-psu-1602.html
- 3D printed parts : 3d printer and filament from Arch. E.C.E. laboratory
- Various Weights : fishing equipment store
- String Roll: equipment store
- Drawing Canvas: https://www.e-jumbo.gr/scholika/zografiki/zografiki-se-kamva/
- Jumper Wires: Arch. E.C.E. laboratory
- Easel/artist's tripod: https://www.hobbywood.gr/colors/kavaleta?product_id=115277

Total cost of the Robot ~=60

Dimensions of prototype

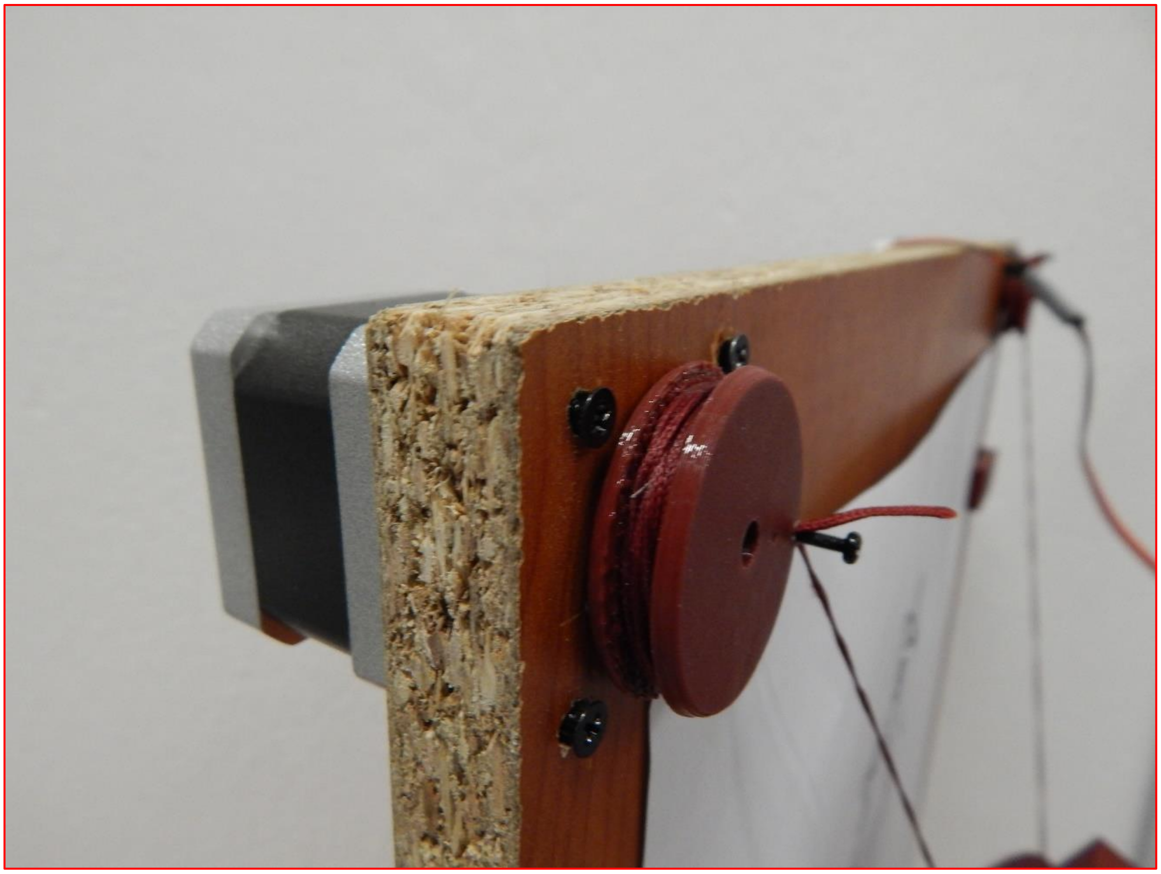
- Machine width:52.7 cm
- Machine height:63.2cm
- 3D printed parts dimensions in the .stl files included with zip

Upgraded Version

Since the first robot was a prototype so as to test, try and debug, a more "clean" and solid version was also built. The experience from the prototype were directly used in the second one as well with some minor hardware changes, which are analyzed further below. The same software is used in both robots.

Hardware changes

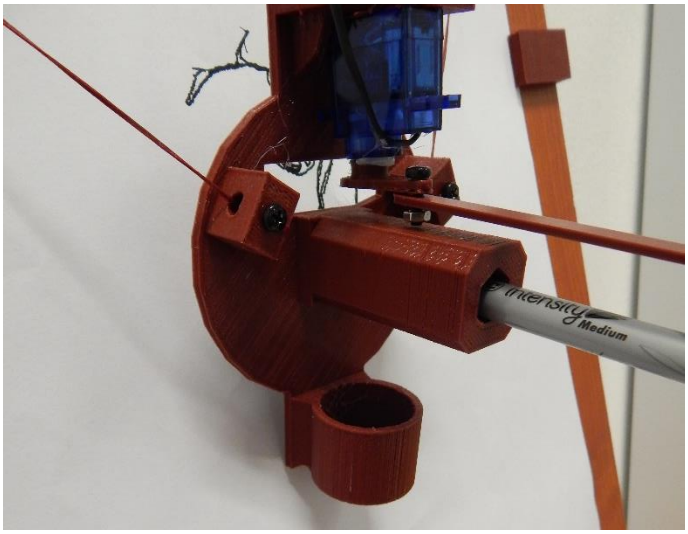
As it is obvious, there were some hardware changes both visual and functional. To begin with, this time, a wooden back was used for more steadiness and solidness. Furthermore, the motors were put in the back of the board for a "cleaner" look and practicality, since the 3D printed motor holders might easily break or malfunction. Holes were drilled, after taking measurements first, so the motors can pass through the wooden back.



Also, the 3D printed pulleys this time were designed for a simple string and not for a beaded string. This made no difference in the performance of the robot but surely made the construction less bulky and easier to repair in case the string is cut.

Gondola Mechanism

The pen’s mechanism on the gondola was redesigned this time. The result made the pen’s performance more steady and more precise that the prototype. The mechanism is nothing less than a 3D designed solution of the prototype’s “DIY” and unsteady one.



Paper holders

A problem with the prototype was that the paper was not standing still and during runs sometimes it was going out of place ruining the result of the drawing. This time some simple paper holders were designed using small magnets. 4 holes were drilled halfway through the wooden back so the small magnets would be installed without obstructing the paper to be flat. The 3D printed paper holders also have one magnet installed in them. This was a simple and really good solution which kept the paper in place nice.

Extra parts

Finally, some extra parts were printed but not necessary for the function of the robot. Specifically, an Arduino case so the Arduino and shield can be installed in the back of the board and a pen holder for extra pens.

The 3D .stl files can be found in the project file

General Tips

In order to achieve good runs, you will need to make many tests to figure out the right settings for your machine. Try to make your values as precise as possible when configuring your machine in the Polagraph GUI. It is expected to have many obstacles concerning mostly the hardware. Try to have as less as possible loose parts, especially the motors and the pulley because it can easily lose steps due to malfunction and ruin the robot's performance. Since in the making of this project there was not a compact place for FAQ's and answers, this documentation provides solutions to many cases for troubleshooting. Further answers can be found in the Polagraph Forum.

Final notes & Tips

- The hardware can get really heated, especially the motors. It is suggested that after the built, some kind of cooling is installed. For this project a future cooling system with fans is going to be installed cause this kind overheating can cause damage to the robot.
- Before buying the components needed, it is suggested to pay attention to the details and give the project a good thought. This can save you from many obstacles in the future.

Sources

- https://www.polargraph.co.uk
- https://www.instructables.com/Polargraph-Drawing-Machine/
- https://www.thingiverse.com/thing:575487
- https://www.youtube.com/watch?v=T0jwdrgVBBc
- https://github.com/euphy/polargraphcontroller/releases/tag/1.10.6a