AxiDraw Pen Plotter Documentation

Studio Documentation February 16, 2021

Meet Harry (the plotter)

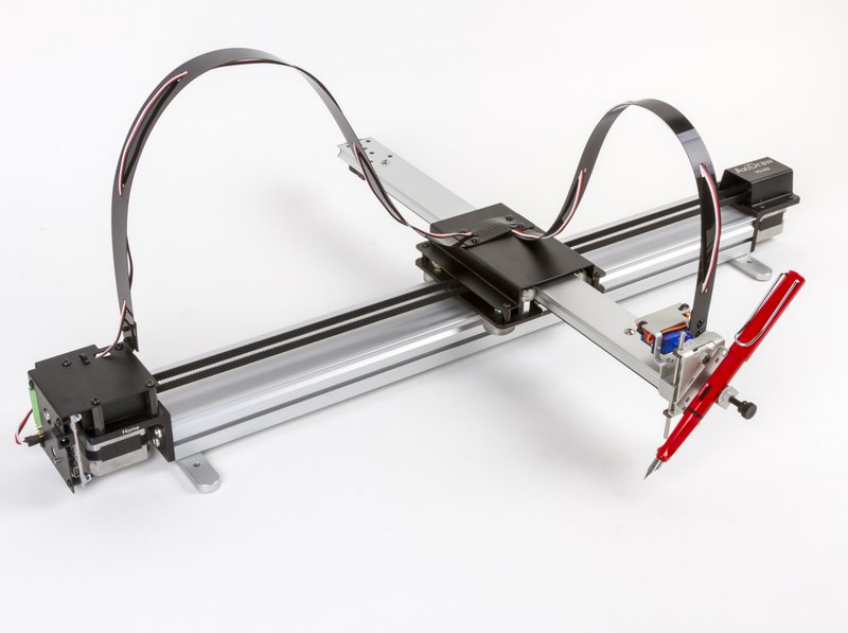

Figure 1: Harry (the plotter) – an AxiDraw V3/A3

The AxiDraw is a simple machine capable of writing on most surfaces, using most kinds of pens. It's entirely computer controllable–which makes it a great (if simplistic) system to experiment with how code can affect the physical world!

These kinds of machines are often called *pen plotters*, and in the early days of computing were frequently used to print high resolution graphics in domains from business to science. They are now mostly replaced by laser and inkjet printers, but have a vibrant following of computer artists and hobbyists. If you research computer-generated graphics from before the 1980s, it was likely printed on a plotter not dissimilar from ours. (In 3Ai there is also a special piece of computer generated art from the 1968 conference *Cybernetic Serendipity*, also printed on a plotter. See if you can find it!)

Set up and installation

AxiDraw set up

To become acquainted with the AxiDraw plotter, begin by reading the User Guide: `https://cdn.evilmadscientist.com/dl/ad/public/AxiDraw_Guide_v45.pdf`, in particular we recommend reading **Chapter 2: Checking out your AxiDraw** and **Chapter 4: Quick start: Making your first plot**. You can also find the user guide PDF on Wattle.

If you wish to generate and render AxiDraw designs using vector graphics (.svg files) such as from Adobe Illustrator or Inkscape, we recommend also reading **Chapter 3: Software for AxiDraw**, and following their instructions for setting up the AxiDraw Inkscape extensions. The computer connected to the AxiDraw also has this software available.

Coding set up

The Build course uses a variant of the open source library called `axi`. To access this, download and unzip the zip file `induction.zip` from the Build Wattle.

Use the instructions below to install the plotting code on the system you will be using for the course:

Windows

1. **Install Anaconda as a package and environment manager for Python.**
   Follow this tutorial. You do not need to install PyCharm.
   `https://docs.anaconda.com/anaconda/install/windows/`

2. **Install Visual Studio Code.**
   Install Visual Studio Code
   `https://code.visualstudio.com/`

3. **Install Python dependencies for AxiDraw**
   In an Anaconda Shell (search for Anaconda prompt in your windows search bar), open the folder where the unzipped `induction.zip` is located, then go to the terminal and type the below commands

   ```
   (base)$ cd induction
   (base)$ conda env create -f environment.yml
   ```

   At this point your build might fail. If you see an error that recommends you to install visual studio C++ build tools, follow the link and just select the Visual Studio Build Tools. At this point, you may need to restart your system, and remove the partially built environment. **Optional:**

   ```
   (base)$ conda remove --name 3ai-plotter --all
   ```

Once you've completed these, try recreating your environment by running the command "`conda env create -f environment.yml`". When complete, run the following.

```
(base)$ conda activate 3ai-plotter
(3ai-plotter)$ pip install -e ./axi-master
```

4. **Test your installation**
   Run the following commands

   ```
   (3ai-plotter)$ python letterA.py
   ```

   This should output an image of (most of) the letter 'A' as a .png

Mac and Ubuntu

1. **Install Python and Visual Studio Code.**
   Download Visual Studio Code off its website: `https://code.visualstudio.com/` to install Python, follow the install instructions at `https://www.python.org/downloads/`, by choosing the version of Python appropriate for your OS.
   Test that Python works on your system by running the command

   ```
   $ python --version
   ```

   The output should be `3.9.1`, or a number close to that.

2. **Install Python Dependencies for AxiDraw**
   Open a terminal in the folder where the unzipped `induction` directory is located. Then let's install the Python dependencies using `pip3`. If you do not have `pip3` installed, you can do so by following the instructions on this page: `https://pip.pypa.io/en/stable/installing/`

   With `pip3` installed, run the following commands:

   ```
   $ cd induction/axi-master
   $ pip3 install -e .
   ```

3. **Test your installation**
   Run the following commands

   ```
   $ cd ..
   $ python3 letterA.py
   ```

   This should output an image of (most of) the letter 'A' as a .png

What else can axi do? You can see examples of the library's usage in `axidraw-master/examples`.

Bonus: The makers of AxiDraw have also released an official Python API called `py_api`. We chose not to use this one for teaching primarily because it does not allow for previewing images of prints, as well as not having a Turtle interface (discussed below). It is, however, very well documented. You can check it out at: `https://axidraw.com/doc/py_api/`

Drawing with Python

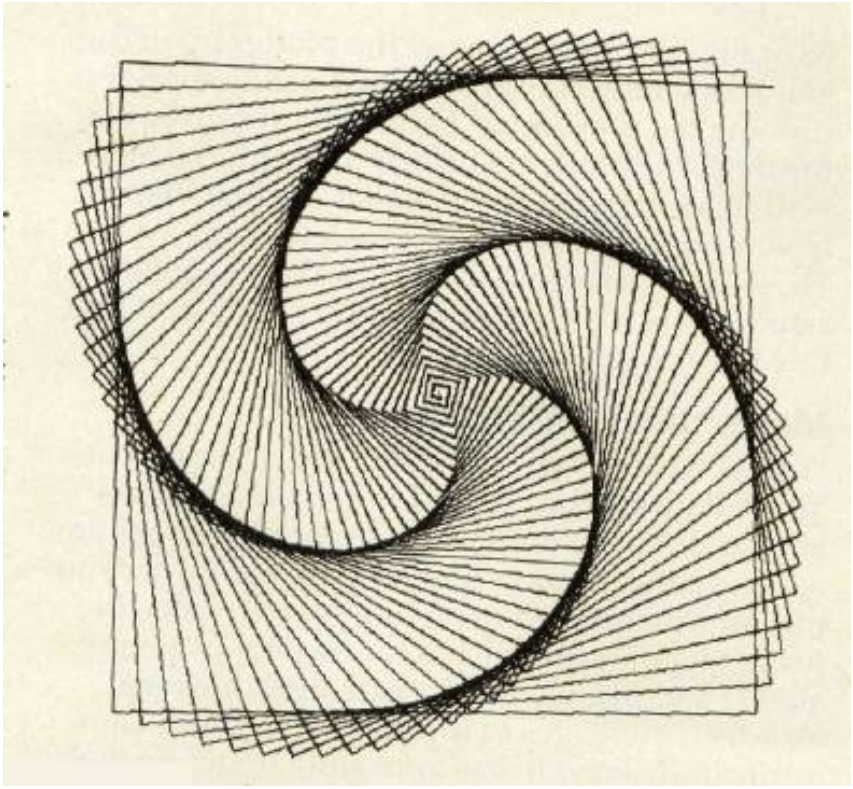

Figure 2: A plotted drawing made using Logo in 1984. Source: COMPUTE! ISSUE 46 / MARCH 1984

To get started Drawing with Python, we introduce the **Turtle** graphics interface. It uses the same set of turtle commands popularized in the Logo programming language by Wally Feurzeig, Seymour Papert and Cynthia Solomon in 1967.

The basic premise is that the turtle has a position, orientation, and pen. The pen itself also has the attributes of being on or off, and its color and line width is determined by the *actual pen* you have in the plotter. This allows you to issue commands like "go forward 10 steps, then rotate 90 degrees left" or "put pen down, then move backwards 5 steps". One main advantage of writing graphics this way is that you can imagine yourself as the turtle. This can be especially handy when you are first starting out. The image above was drawn using a set of turtle commands. Can you imagine what they might be?

Task 0: Tinkering with (virtual) Turtle

To get a sense of how Turtle works in Python, you can use this online interactive tool:
`http://www.pythonsandbox.com/turtle` *(Note, the units here are in pixels, whereas for the plotter they are in inches, so your code won't transfer directly.)*
You can view which commands are available in the Python documentation:
`https://docs.python.org/3/library/turtle.html`

Once you are familiar with the basic operations for creating drawings, like **`forward`**,

penup, pendown, right, left, goto, go to the next section to learn how to create drawings on *real paper* using Turtle.

Task 1: Plotting with Turtle

Below is the beginning of some code to draw the letter 'A' using Turtle. We encourage you to follow along, and by the end we hope you will have your very own A4-sized letter A! Having drawn this letter will be necessary to be inducted to use the Pen Plotter.

The code: Below is the snippet of code needed to draw the letter A. Give it a read, then go to the `examples` directory. The same code below is saved in the file "**`letterA.py`**"

Source Code 1: Sample Python code to plot letter A

```
import axi
# Set this variable to true when you wish to draw with the plotter.
draw = False

def main():
    turtle = axi.Turtle()
    turtle.pd()
    turtle.goto(2, -5)
    turtle.goto(4, 0)
    turtle.pu()
    turtle.home()
    # Scale letter to fit A4-sized paper
    drawing = turtle.drawing.rotate_and_scale_to_fit(8.5, 11.75, step=90)
    # Render drawing
    im = drawing.render()
    # Create an image file of the drawing
    im.write_to_png('out.png')
    # Draw with the plotter if relevant
    if draw:
        axi.draw(drawing)

if __name__ == '__main__':
    main()
```

Run the above file, as is, to generate a preview image of the drawing. The preview image will be called `out.png`

```
$ python3 letterA.py
```

Open the output image. It should look like this:

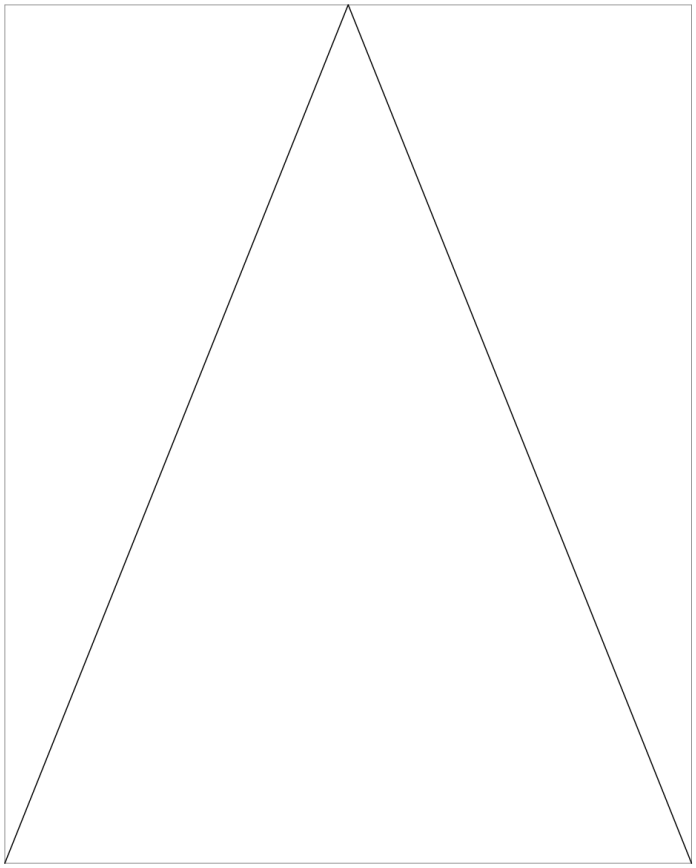

Figure 3: Output of the program `letterA.py`. The grey lines are the border of the A4 paper.

But Wait! This isn't a proper A! The line going across the middle is missing. Your first task is to modify the Python file `letterA.py` so that it outputs the image of the letter A.

Task 2: Getting physical – Plotting with the pen plotter

Now that you've created your own letter 'A', it's time to plot it using the pen plotter. Copy it on a USB stick, and ask Harry to plot it for you. Instructions on how to make your first plot are detailed in **Chapter 4: Quick start: Making your first plot**.

When you reach section 4.7, open a terminal in the directory where your Python code is copied and run the code. Be sure to change the value of variable `draw` from `False` to `True`.

Appendix I: Drawing with vector graphics

Vector graphics are drawings defined by mathematical formulas. They have special properties, like being able to be scaled without distortion. Many vector graphics are used in the `.svg` format. You can draw vector graphics with the pen plotter using either the official penplotter command line tool, `axicli` or InkScape, a popular open source vector graphics tool.
For installation instructions, consult **Chapter 3. Software for AxiDraw** in the User Guide.

Appendix II: Troubleshooting

Starting out a print

If you skipped reading the user guide, go back and read it! In particular **Chapter 4.** But over all, the guide is really helpful, and will show you the ins and outs of running a successful plot. Please do not attempt to print until you have read these instructions.

As a refresher, to start a print:

1. Attach a sheet of paper, or whiteboard to the **clip easel**.
2. Turn off plotter, and return the carriage to **home-corner**, pusing only on the **safe holds** (cf. user guide Ch 4).
3. Connect the power and USB. Likely, you will need to engage the switch on the wall.
4. Toggle the pen to the UP position using Inkscape (Ch 4. of the user guide instructions how to do this).
5. Insert your pen. Be sure to properly prime it so that the ink flows. Be selective in which pens you use–felt tips and roller balls (but not ball points) work well.
6. Position the paper so that it fits precisely under the pen.

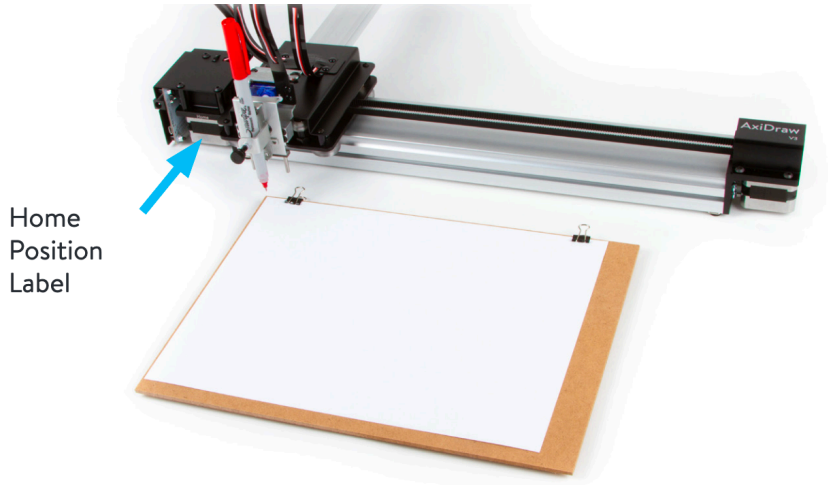

Figure 4: The plotter in the home corner position. Do not move the carriage without shutting off the plotter. Consult the user guide for specific instructions.

You made a mistake! Stopping the plotter mid-run.

Sometimes you may find that something went wrong with the print, and you would like to stop and reset. To do so, either press the Pause button in Inkscape, the pause button on the side of the plotter (if Inkscape is running the print), or otherwise you can kill the Python script running the plot job, such as with `ctrl-C`. If you choose this option, please ensure that the

pen is returned to the UP position. You can do so either using the Inkscape extension (see Ch. 4 of the User Guide), or by using the `axicli` command-line utility:

Toggle the position of the pen. So if it is down, it will go up. And if up, down.

```
$ axicli --mode toggle
```

To return to home corner, you should either turn off the AxiDraw or you can also use `axicli` to disable the xy-motors:

```
$ axicli -m manual -M disable_xy
```

To enable the motors:

```
$ axicli -m manual -M enable_xy
```

See the remaining axicli options here: `https://axidraw.com/doc/cli_api/#manual_cmd`