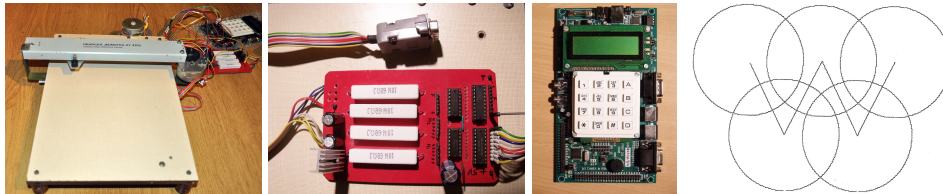


A Low-cost Plotter Controlled by FITkit

Ivan Ševčík*



Abstract

This paper proposes a cheap and customizable device for drawing vector graphics on variety of surfaces. This includes creating a drawing hardware, a control unit and its software. The drawing device was designed as a plotter with two axes of freedom, allowing a drawing surface to remain static. An embedded platform FITkit was used as the control unit in order to save money and reduce time spent on the project. The FITkit had to be accompanied by an amplifier board which bridges its output ports and the motors, as these operate at a different voltage levels. A software for FITkit was developed that communicates with a personal computer and converts vector graphics to the motor movements and pen control. Finally, a dedicated program was developed also for the computer side. It provides a convenient way to issue commands and read files containing vector graphics created by a suitable editor, such as AutoCAD. The result is a fully functional and highly customizable plotter-like device that can draw both individual commands and whole AutoCAD DXF files. This work may serve as a starting point for creating a drawing device like the one presented in this paper. Having such device available opens up a lot of possibilities and it represents a valuable tool for future projects.

Keywords: FITkit — Plotter — XY 4150

Supplementary Material: [Demonstration Video](#) — [Downloadable Code](#)

*xsevc50@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

1. Introduction

It is common that data stored in a personal computer (PC) need to be converted into some physical form. There are many ways how to do that but probably the most common one is by printing on paper using printer. However, sometimes it is needed to be able to draw on other surfaces, such as plastic, or use different method instead of drawing. Conventional printers are not good at this. One of the solutions is to first print on paper and then tape it onto a desired surface. Another solution is to build a new plotter-like device. It is characterized by a head that moves a tool over the surface, in contrast to moving the surface through the device. The benefits in-

clude a wider range of material thickness, elimination of problems with gripping and moving small or heavy objects, and preventing smudges when drawing. The type of the tool may vary from pens to laser diodes. Such device can be used for a variety of purposes, for example to make glass paintings, mark cutting lines on plywood or prepare printed circuit boards. After equipping the device with a laser diode instead of the pen, some surfaces can be cut directly while others can be at least engraved, depending on a power of the laser diode. This could be used by hobbyists to cut out parts of a plane from plywood, artists to create drawings on wood, or to cut out various masks and templates.

The devices of this type already exist in some form

but are expensive and hard to modify without losing a warranty, making them one-purpose only products. The goal of this work is to construct a low-cost and customizable version in a spirit of a do-it-yourself (DIY) project. The device should be able to draw, or more generally plot, at least basic geometric primitives, such as lines, circles, and arcs without moving the drawing surface itself. The solution should also include a unit that would control the plotting process and communicate with the PC. FITkit, an embedded platform, could be used for that purpose. Finally, the solution should provide a simple and convenient way to process some popular file formats containing vector graphics.

There are some other DIY projects that provide plenty of information and inspiration. However, none of these use FITkit as the control unit. Therefore, building them would either require buying another development platform, like Arduino, or to create a custom control unit from scratch. From the construction viewpoint, a good quality project is XY Plotter [1], which seems to have a very stable and vibration-free construction, and can draw shapes reasonably fast. It is constructed from parts of a construction kit similar to Merkur [2], which is popular in the Central Europe and might be used as a substitute. Interesting is the choice of a servomotor for controlling a pen level, as it may be later used to implement different amounts of pressure. Another project that is well-documented, has a public source code, and even shows how it can be upgraded with a laser diode, is Motöri [3]. It also serves as a good example of a low-cost solution as it uses salvaged parts from an old printer.

Some components from an older hardware, such as stepper motors and a part of a graphic unit, were used to construct the drawing device presented in this paper. As noted earlier, FITkit was used as the control unit. Because FITkit doesn't have a necessary output power to run the stepper motors on its own, an amplifier board was needed to bridge the voltage differences. Finally, both a FITkit and a PC software were developed. The former controls the motor movements and implements drawing algorithms that transform vector graphics into specific movements. The program for the PC side provides a console for user input and implements a queue for processing the drawing commands.

The result is a fully functional device that is simple enough to be made at home for a reasonable price. The actual cost depends on which parts are already available but even when having none of them to start with, the price will be a fraction of the commercial solutions. With the software source codes and schematics made public, this is almost an out of the box solution for any-

one who has access to FITkit. However, the code was written in such way that adapting it to another platform should require only minor changes in communication and port control.

2. Information about Utilized Platforms

In order to save money and time, this work builds upon some already existing platforms. The sections below will provide more details about them. It is worth noting that they were chosen because of their availability, not popularity or accessibility. Therefore, it is probable that when someone decides to build their own device, something will have to be substituted. This is common with DIY projects and this work assumes some level of skills so that one can make the required changes. However, it also tries to make the process of substituting as easy as possible.

2.1 FITkit

FITkit is an embedded educational platform available to every student of FIT BUT. Currently at version 2.0, it features a fast micro-controller (MCU) belonging to a MSP430 family from Texas Instruments, and an FPGA Spartan 3 from Xilinx. FITkit is programmed via a built-in USB controller FT2232C which also provides a serial communication channel. Some pins of the MCU are connected to a pin header so that custom peripherals can be connected [4].

There are two libraries available that can be used in the projects involving FITkit. First one, `libfitkit`, is used by programs that are executed in the FITkit's MCU and provides basic functions to programmers, such as serial communication with a PC over USB, FPGA initialization and configuration, a watchdog, and others. The second library, `libkitclient`, is intended for use by PC programs to find, list, and communicate with the connected FITkit devices. A dedicated application called QDevKit is also available to developers. It assists with code compilation and MCU flashing, and provides a simple terminal window.

2.2 Graphic Unit XY 4150

The XY 4150, pictured in Figure 1, is an old Czech plotter [5] from 1989 that was originally controlled through parallel port. Connecting it to a modern PC would require a serial to parallel converter. Originally, three wires determined a selected motor, a direction, and a pen position, and the fourth wire actually signaled a movement. The computer was involved in the whole drawing process all the time, controlling each step of the plotter. The other issue was its construction that moved the drawing surface through the device.

Instead of adjusting the plotter for our needs, it was disassembled, providing a drawing head with a pen control mechanism based on a coil and two stepper motors [6] with step resolution of 4.5° .



Figure 1. The original XY 4150

3. Proposed Solution for a Low-cost Plotter

The proposed solution can be divided into four individual projects. A system diagram in Figure 2 displays their role in the solution and how they interact.

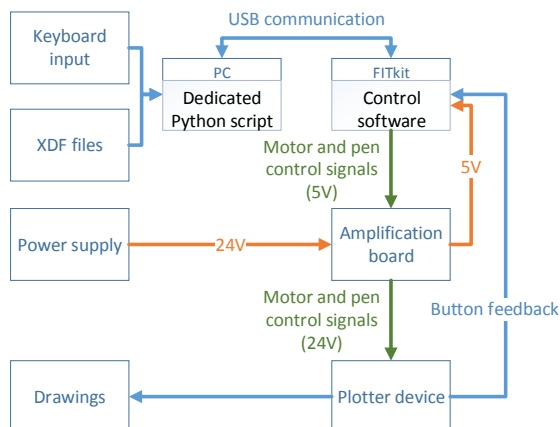


Figure 2. The solution system diagram

3.1 Design of Plotter Hardware

The design is inspired by CNC routers and laser cutters that are moving a head with a tool over a surface that remains static. A wooden board that supports the drawing surface also creates a base of the device. The legs are mounted to the bottom of the board, providing some room for the shafts and a mechanism that will move the graphic unit in a perpendicular direction, in this case vertically. The Figure 3 displays this system as a whole. The graphic unit is fixed to a carriage which is mounted to one of the shafts, while the opposite side is just resting on the second shaft. This can be seen in a detail in Figure 4 and it prevents the carriage from getting stuck due to imperfections in the construction. The moving mechanism consists of the stepper motor with a gearbox that pulls a string connected to

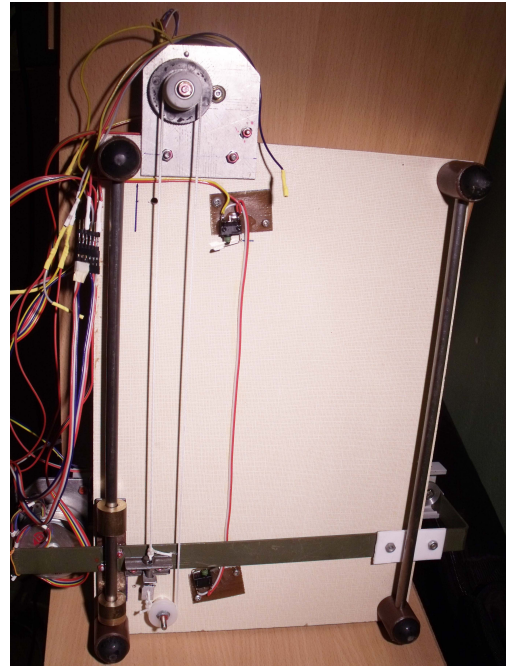


Figure 3. An overview of the plotter's bottom side

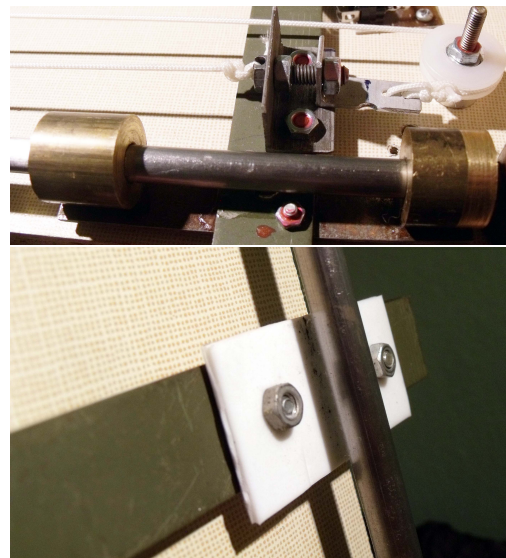


Figure 4. The shafts and the carriage

the carriage. The graphic unit has a similar mechanism for moving the drawing head, which is displayed at Figure 5. By using the gearboxes, a drawing resolution of approximately 0.1 mm was achieved on both axes. Finally, the pen is fitted in a screw and mounted into the drawing head that contains a little coil and a spring. When a current flows through the coil, it pulls a small magnet, putting the pen down. After the current stops flowing, the spring retracts the pen back into an upward position. To prevent a damage of the hardware, buttons have been placed on both axes in each direction. They are tested in the software for a press and stop further movement in that direction. Using the buttons to designate a drawing area has also an advantage that the size of it doesn't have to be configured anywhere.

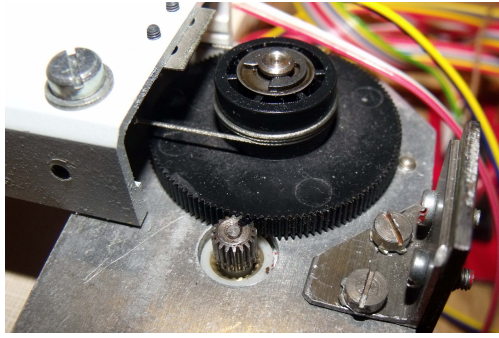


Figure 5. The gearbox of graphic unit with attached string that moves the drawing head

Simply moving the buttons further apart would also increase the drawing area without any need of software changes. The button pressed by carriage is shown at Figure 6. There are some disadvantages to this simple

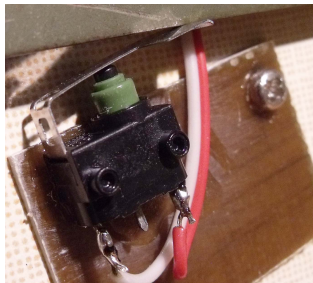


Figure 6. The button that designates the drawing area pressed by carriage

plotter design, such as a low tolerance to vibrations originating from motor movements. Another, related problem is that moving the whole carriage with the graphic unit, which is relatively heavy, results in a lot of inertia that limits the drawing speed.

3.2 Amplifier Board

An amplifier board between FITkit and the stepper motors is needed to bridge the voltage differences and the current requirements. This would be required by any similar development platform because output pins are generally directly connected to an MCU, which is running at 5 V and can't supply enough current. Therefore, a custom amplifier board, shown at Figure 7, was designed and constructed for purposes of this project. It is an electrical circuit that involves resistors, darlington transistor arrays and bus transceivers. The amplifier board is powered by a power supply that provides 24 V required by the motors.

3.3 FITkit Control Software

The control software is divided into four parts. The first is a finite-state machine that represents the body of the program. Because there must be a short delay between motor movements, which is determined by

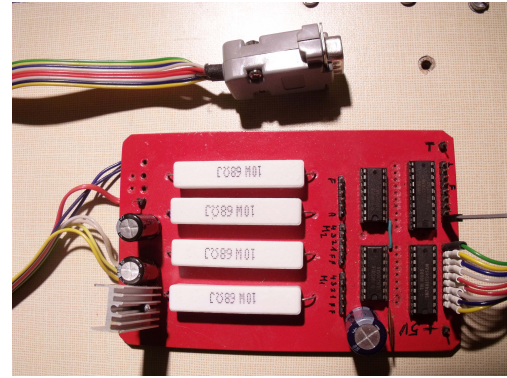


Figure 7. The amplifier board

physical limitations, drawing a whole shape as a single operation would make FITkit unresponsive. This is solved by defining a state and a context for each drawing and then executing one step of an algorithm at a time. The second part implements the drawing algorithms. The currently supported shapes are lines, circles and arcs. The original intent was to use well established rasterisation algorithms but some modifications had to be done. For example, the *circle midpoint algorithm* [7] incorporates symmetry, so that only one eighth of a circle needs to be rasterised. However, it would be very inefficient to move the pen between eight positions all the time and also the result would be very poor. Therefore, to make the pen movement continuous, the whole circle needs to be rasterised. This is also true for arcs as they use the same rasterisation algorithm. In order to rasterise lines, an original *Bresenham's line algorithm* was used [8]. The third part consists of the motor movement routines. The motor is turned one step by switching its coils on and off in a correct order [6]. This translates in the program to sending series of words to the output ports. Finally, the program implements a command interpreter which identifies the commands received over USB, extracts the parameters, and issues the drawing calls.

3.4 System for Communication and User Input

A dedicated Python script was created to make the input of the commands more convenient. It also implements a queue so that the commands are kept in a memory while a current shape is being drawn. The most important feature is a support for DXF files, allowing to use a visual editor to draw shapes and then use that file as an input for the plotter.

4. Demonstration of Results

The results of our project are presented in Figure 8. The upper part shows an output of a complex built-in command `HILBERT`, which draws a Hilbert curve

of a selected degree into an area of $15\text{ cm} \times 15\text{ cm}$. The graphics in the lower part were first drawn using AutoCAD and then the file was used as an input. The output of a drawing process is not perfectly precise due to some vibrations of the plotter construction but should be good enough for the intended hobby use. Some interesting plotter parameters are summarized by Table 1.

The design was to a large extent influenced by the available components. This is a common practice with DIY projects. In our case the project was completed spending under 10 €. The software for FITkit and PC, schemes, and other files are available under MIT license at GitHub [9].

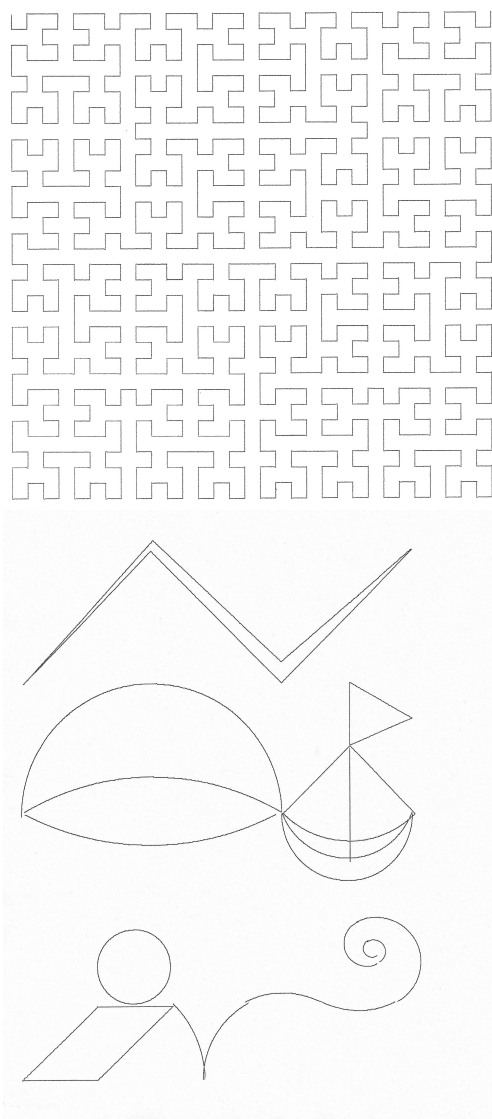


Figure 8. The drawings created by plotter

5. Conclusions

A plotter is a very versatile device that can be used to draw on a variety of surfaces, using a wide range of pen types or even other tools. This work aimed to cre-

Supported surface size	A4 paper
Step resolution [x,y]	[0.1, 0.12] mm
Maximum speed	250 steps / s

Table 1. Plotter parameters

ate a simple plotter operated by FITkit. We show here that it is possible to achieve the goal without spending lot of money, using parts that might be considered obsolete. This work adds up to a collection of FITkit projects. Other students can use it to learn how to control the stepper motors using FITkit. It also provides schematics and source codes for anyone who might have XY 4150 or similar available and is looking for a way to make it usable again. The plotter will be used as a helpful tool in other projects and will be improved over the course of time. The plans also include a purchase and experimentation with a laser diode.

References

- [1] Xy-plotter. <http://www.contraptor.org/xy-plotter>.
- [2] Merkur. <http://www.merkurtoys.cz/>.
- [3] Viacheslav Slavinsky. Motori the plotter. <http://svo.2.staticpublic.s3-website-us-east-1.amazonaws.com/motori/>.
- [4] Fitkit. <http://merlin.fit.vutbr.cz/FITkit/docs/en/uvod/uvod.html>.
- [5] Technical documentation for xy-4150 [cz]. http://xy4150.webstones.cz/dwl/XY4150_Puvodni_Technicka_Dokumentace.pdf.
- [6] Smr 300-100 stepper motor. http://www.electromotor-bor.com/KM/km_300-100ri.html.
- [7] Circle midpoint algorithm. <http://ezekiel.vancouver.wsu.edu/~cs442/archive/lectures/raster/circrect/circrect.ppt>.
- [8] Bresenham line algorithm. <http://www.cs.helsinki.fi/group/goa/mallinnus/lines/bresenh.html>.
- [9] Project repository. <https://github.com/BetaRavener/FITkit-plotter>.