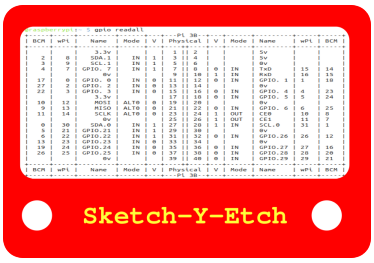
Sketch-Y-Etch a retrospective drawing experience for the masses

SensoredHacker0 @ The Bangor Makerspace

March 19, 2024

Abstract

The Sketch-Y-Etch is a pseudo clone of the classic Etch-A-Sketch with a modern twist. Features include similar Cartesian drawing using knobs, with the addition of being able to change colors, lift the pen, erase segments, Erasing without shaking, saving files, and its huge. Written in python, utilizing standard hardware, and some open source projects, lets explore the Sketch-Y-Etch, and how you can create your very own.



History

The Sketch-Y-Etch is thematically based on the classic Etch-A-Sketch. The Etch-A-Sketch is a mechanical drawing toy invented in 1960 by André Cassagnes of France and subsequently manufactured by the Ohio Art Company. Similar to the original, knobs are used to control x,y axis drawing. Sketch-Y-Etch however is a computerized project, and similarities end there.

Programming:

The Sketch-Y-Etch program uses a micro controller acting as a keyboard, and Python for graphics. The Python3 code listens for key commands, and the micro controller provides them via IO. A standard keyboard would also work. Source Code is available at:
https://github.com/FOSSBOSS/SketchyEtch

Software Resources Used:

OpenScad to 3D model the Knobs.
The micro controller interface was coded in Arduino IDE using C++.
The drawing program is in Python3, using the turtle library.
Unclutter is used to hide the mouse cursor.
Nginx is used to provide a web interface to retrieve saved files.

Construction:

The Sketch-Y-Etch project is designed to run on standard computer hardware. Making A version using buttons and knobs is entirely optional, though in our opinion, much cooler.

Material needs: Two rotary encoders, four buttons. a micro controller capable of acting as a keyboard, a computer of some sort, and a TV.

This project utilizes a raspberry Pi 3b+, and a Samsung ln26A330j1 30 inch TV. Any TV, and computer of your choice should work. This model TV, has wide bevels, and lots of internal space where hardware can be embedded. If you can not get a TV with space to embed various additional electronics, as many modern TVs do not have bevels, you could create a control panel, or make a box to contain your additional electronics.

Tribulations of the build:

This project went through many revisions. The initial version had a 6 by 8 foot screen and stood nearly 10 feet tall. It was also heavy, and impossible to move.

Raspberry Pies have GPIO.. why did we not use that for IO? Several versions were created to utilize Raspberry PI GPIO. Testing revealed latency problems, that we did not want to deal with. Also, utilizing standard hardware maximizes the number of devices, and subsequent variants which can be built.

The second revision implemented the idea of using keystrokes to control navigation. This allowed us to share the Sketch-Y-Etch concept with other developers interested in the project, without forcing them to have a Raspberry PI, and custom electronics.

Operation:

Usage should be intuitively similar to using an Etch-A-Sketch. Turn knobs, make pictures. Cycle through a set of colors with a push button. Momentarily stop drawing, and move your marker with the lift button. Erase it all with the

erase button. Save an image using the save button, and for display purposes, use the demo switch to run a demonstration mode which showcases various drawings, documentation, and other info.

To use the Sketch-Y-Etch drawing program with a keyboard, use these **left** and **right** arrow keys to navigate on the +/-X plane. **Up** and **Down** arrow keys navigate the +/- Y plane. Press **g** to change the color. Pressing **g** again will shift to the next color in the list. pressing **g** until the white color is selected enable the erase function, as the background is white. press **l** to toggle the lift pen function. Lifting the pen allows the cursor to be moved, without drawing. Press **c** to clear the screen. The cursor will reset to the center position, and erase everything which has been drawn. Press **s** to save the file, and **d** to enable demo mode.

Annex of caveats and deviations:

The rotary encoders in use with the Sketch-Y-Etch were chosen based on availability. The encoders feature detents, which in this application, have a tendency to cause the encoder to shift forward, or backward when a user stops on the ridge of a detent. Unexpected drawing may occur. Perhaps we could add something in code to compensate this action, or maybe next time we'll use a more apt encoder type.

Latency is built into the sensor IO board, to prevent key spamming. Additional latency is added to the save key, to deter excessive saving of images.

The Python script may require modification to save at an appropriate location.

Annex of variations:

The Sketch-Y-Etch was developed on a variety of Raspberry Pies, and an assortment of TVs & monitors. We chose a Raspberry PI 3b+ as a minimum operational specification. Should you build your own version, and encounter glitches, or various nuances, **The Bangor Makerspace** is certainly interested in offering assistance. The Sketch-Y-Etch was developed 100% on Linux, and has no other platform testing to date. Given how the Sketch-Y-Etch program operates, we see no specific reason it can not be adapted to any other platform; we simply haven't tested this.

Conclusion

The Sketch-Y-Etch is an open source project developed at the Bangor Makerspace. Our intent is to create something fun, and easy to use for everyone to enjoy.