

Coffee drip printer;
Ted Kinsman Assistant Professor RIT
February 2016

For many years I have thought about building a machine that could paint for me. I went through a number of ideas before coming up with the simplest design to build, as well as relatively inexpensive. A drip printer allowed me to experiment with a number of variables and I decided on coffee to be a user-friendly medium.

The machine allows experimentation with drip height, drip size, drip chemistry, spacing of drips, and especially the paper that the drips fall on – all of these affect the image results. I will spend some time talking about each of these in turn.

The microcontroller is an arduino which can store about 32Kb of memory. This allows the microcontroller to store an image of approximately 80 by 100 pixels. Surprisingly, this is enough data for a human to recognize a face, while other subjects often appear random. The process for making an image can be broken into a number of steps. An image is taken at very low resolution and is curve adjusted to make the darks a bit darker. Typically a jpg will be stored and converted to a pgm gray scale file. This file in turn is patched into the arduino ino file with a text editor. (see example code). The file stores the image as an array in which each position on the x-y table corresponds to a location in the array. The pixel is stored as a value from 0 (no coffee printed) to 256 which is defined as the largest drip size. To convert the pixel in coffee, the pixel value is converted to time. In reality, a valve open for .256 seconds creates too large a drip, so I divide the time by a correction factor for example, 256 divided by 4 lets the valve open for 63 milliseconds to create the largest drip, ultimately will become the darkest pixel on the image. There are other unique problems along the way – for instance, the valve takes 10 milliseconds to open, therefore any value corresponding to a lighter pixel is just not printed. This current setup yields about 53 different shades of coffee. Bigger paper can print a larger image having larger drips with more shades, but for the most part this number works fine. The second variable that needs to be calibrated is the spacing between drips. This is a function of the steps required for the stepping motor to make one revolution and the diameter of the pulley on the stepper. In my setup, a value of about 90 steps is required so that the drips do not touch each other when they are wet. The drips get smaller when they dry so the resulting size of the coffee spot is a function of the wetting angle. The wetting angle is a result of the chemistry of the coffee interacting with the chemistry of the paper. By changing the wetting angle a number of unique effects can be created.

The wetting angle ultimately limits how close a drip can be placed to the next drip. The coffee printer currently has a resolution of about 4 drips per inch. This value can be driven a bit higher, but the arduino can only hold so much memory. The limit in image size is about 80 x 100 pixels. A higher resolution output will make a

smaller image – leading to a smaller drip size which in turn will lead to a smaller number of defined levels of coffee darkness that can be printed.

To control the drip valve, a pin on the arduino goes high and drives a reed switch which opens the valve. The reed switch is very inexpensive, and makes a little chirping sound so I know it is working. The sound turns out to be a great indicator to determine if the coffee flow valve is clogged. The valve is an inexpensive 12 v DC valve but works well when run at 9 volts DC. The valves are actually used all over in industry and are found for a few dollars on surplus websites.

The stepping motors are surplus and require 1.2 amps at 12 volts to run. The controller for the stepping motors is the motor shield from adafruit that is also limited to 1.2 amps. This shield only allows one stepper motor to run at a time. This is not much of a limitation with this device since dripping coffee is so slow to begin with.

The drip valve has the flow limited by running the coffee through a micro pipette which restricts the flow. These nozzles were obtained from surplus lab equipment. It should be noted, the nozzle is a fraction of a mm and the coffee that runs into the reservoir needs to be filtered to prevent particles of coffee from clogging the nozzle. The coffee reservoir also travels with the moving head, which is not an ideal design since this mass can affect the x-y printer location.

The depth of the coffee in the reservoir will affect the pressure and thus the drip size, so a Mariotte' siphon is used. A Mariotte siphon is a bottle that allows air to flow in at a certain level of the reservoir – it allows fluid to be pulled from the reservoir and keeps the pressure very constant. I use a modified sports drinking bottle for this purpose. Keep in mind that each print takes a few hundred ml of coffee and the reservoir should be filled before each print.

Each print takes about an hour to print out and should not be moved until the coffee has time to dry – a process that takes about a day.

Why coffee? Since I always have left over coffee, I thought it would be a fun medium to play with. Just about everyone can relate to coffee and this medium is often used to get people interested in what the machine can do. The printer can really print any liquid. I have printed India ink, watercolors, wine, and even beer. Carbonated liquids like beer cause bubbles when run through the print valve and the resulting print will just be a number of random drips all the same size. It is best to allow carbonated beverages to sit a few days to remove the bubbles.

Before a large print is made with a new paper, the machine needs to be calibrated by setting the distance between each drip. To do this I, run a 16 by 16 drip test pattern. The output image allows me to adjust the height of the dripper and make sure all the components are running well.

During the printing process the paper will warp and develop bulges that can be several centimeters high. This warping of the paper is due to the large amount of liquid in the drips – the nozzle of the dripper needs to be adjusted in height so that it never touches the warping paper. As far as paper is concerned, I use the cheapest good paper I can find - Stonehedge AP275-110 paper is 22 inch by 30 inch and is about \$2.50 a sheet at the campus store at Rochester Institute of Technology (RIT), where I teach. This paper gives good results, and if the print turns out well, it is suitable for display. I strongly suspect that coffee used as an artistic medium is not archival.

The coffee printer has made a number of appearances this past year. It achieved a best of show at the Toronto Maker Faire in November of 2014. It was also shown printing at the Buffalo Make faire in June of 2015 and the Rochester NY Make faire in November 2016. In the fall of 2015 the coffee prints were displayed in Artprize in Grand Rapids MI. In may of 2016 the coffee printer will be shown at the Imagine RIT festival.

Future upgrades.

To increase the positioning of the x-y table, a second drive cable should be placed on the far right of the table. Two steppers would drive the head steadier and keep the rail from binding due to the mass of the coffee reservoir. A builder has to keep in mind that the outcome of the project is to build something fun and not re-invent the ink jet printer.

It would be interesting to have a syringe driven by a stepping motor that can print very viscous materials like acrylic paint, or even a food. The material to choose from would be endless with a print head like this.

This limits the use of the steppers to only one at a time. If I could drive two at a time I would be doing coffee drip Spirographs. Maybe this will be the next project.

Conclusions:

The coffee printer is a fun way to get people interested in what can be done with a small microcontroller like the arduino. Kids and adults all enjoy the dripping action, and watch in awe as a picture materializes from just drips.

Special thanks to my brother Andy Kinsman, for his extreme skill in developing and refining the code in the arduino microcontroller.