Table Stitch: Final Prototype

Sophia Goreczky
Cornell University
Ithaca, New York 14853
sag238@cornell.edu

Katherine Verbeck
Cornell University
Ithaca, New York 14853
kjv26@cornell.edu

Andrew Wisnieff
Cornell University
Ithaca, New York 14853
arw87@cornell.edu

Introduction

Individuals interested in sewing or embroidery may have come across the desire to stitch an image—a shape, a tricky pattern—onto a piece of fabric. This can easily be done sewing by hand but that comes with following the image exactly, relying on human skills, and having to thread the needle by oneself. Choosing to pivot the fabric on a sewing machine into the form of a design can be difficult as well, because the user has to figure out exactly when to pivot the fabric, and if he or she does it too slowly or at the wrong angle, his or her design will be instantly ruined. There is currently a way to stitch complicated designs by buying an expensive embroidery machine, but then it costs several hundred dollars and generally comes with pre-made (or hard-to-create) designs.

To stitch complicated designs on fabric, many individuals must opt for purchasing an embroidery machine, which are significantly expensive. The Table Stitch is a technology that turns a regular sewing machine into a simple and much cheaper embroidery machine. Also unlike the embroidery machine, our Processing interface makes it easy for the user to design his or her own sketches for the machine to stitch.

The Table Stitch moves based on the desired shape, resulting in the stitched image. The basic design goals of the Table Stitch are:

1. The user can select a predefined pattern (a circle, star, and the word "*info*") or draw on the interface a shape or pattern out of a continuous line.
2. The device allows for the fabric and thread of the user's choosing given that the fabric allows for stitching on their particular sewing machine.
3. The device keeps the fabric taut during the stitching so as to prevent bunching.
4. The user manually starts the sewing machine by pressing down on the peddle when the Table Stitch machine begins.
5. The device moves on an X and Y plane to generate the pattern.

The system is very much like a very simple embroidery machine, where the user selects the pattern and the sewing-machine-turned-embroidery-machine will stitch the pattern on its own.

There are some assumptions made: First, a sewing machine will be provided. Second, before using Table Stitch, the sewing machine has been threaded by the user (both the bobbin and needle thread). Third, that before each stitch session begins, the stepper motors are aligned to an appropriate position on the gear racks, because the stepper motors do not know the limitations set by the MicroRAX layer sizes. Lastly, it is assumed that the fabric is loaded before the Table Stitch begins its sequence of movements.

The basic predefined shapes will consist of a circle, a star, and the word "*info*". These can be selected by entering "c", "s", or "l" in the Processing interface created for the Table Stitch. The user can also select points on the interface and clear the points if they wish to start over their sketch.

Related Work

The two basic works that we are building off of are, as aforementioned, the sewing machine and the embroidery machine. The Table Stitch takes inspiration from the embroidery machine and makes changes to the sewing machine to make it so that the sewing machine can stitch shapes or drawings on its own. Other related do-it-yourself projects exist on the Internet, where designs are drawn, engraved, or cut in a similar manner to our machine.

Sewing machines require that the user turn the machine on, using a pedal to automatically stitch fabric. They require the user to move the fabric along in a line or to pivot it, else the stitches bunch up and damage the fabric. Some pricier modern sewing machines allow you to upload various types of stitches, as well as save your own new stitches (ie zig-zag or decorative stitches). To make a stitch, the needle must be threaded, and a bobbin (either top or front-loading) must be set in the machine. First, the needle of the sewing machine comes down from the top and hooks to the bobbin. Then the hook of the bobbin pulls the needle thread over the bobbin and pulls the bobbin thread in the needle loop up, creating a stitch.

Each loop of the needle thread makes another stitch with the bobbin thread [10]. The main issue with the sewing machine remains that it requires manual moving and turning of the fabric [6]. Table Stitch adds to the sewing machine's basic stitching and aims to solve this problem by being able to calculate turning and moving the fabric into the user's desired shape or design.

Embroidery machines are expensive machines that take in embroidery designs and stitch it out for you. However, most embroidery machines require a certain format of designs that can be purchased or sometimes get for free with the program or online. You can load new images into your embroidery program, but it takes a lot of work to make the image both readable and capable of being stitched correctly [2]. The Table Stitch will be something that can stitch easy, personalized designs, cheaply. With embroidery machines, all the movement is in the needle, since embroidery requires that the fabric stay tight or else the design will not come out as planned. With the Table Stitch we need to make sure that the fabric is tight, but that the fabric still moves and rotates on its own; since the designs are much easier than actual embroidery it should be possible.

Machines designed to guide the movement of a pen have been the subject of at least one do-it-yourself project [3]. Automating the process of drawing on paper is already highly relevant to our project; the only difference is that we want a sewing machine to interact with the surface instead of a pen. Granted, one big difference here is that the instrument moves instead of the material; in this project's case, it would be wise to move the material considering the fact that a sewing machine would not be so easily moved in this manner. One particularly useful feature we learn from this video is that the user should have the ability to move the material outside of programmed procedures (as demonstrated at 1:24 in the video). As this hobbyist demonstrates, this can prove to be a very simply way to "line up" material prior to activating the programmed sequence. Extremely similar implementations exist across the Internet, including ones that use lasers to engrave designs [8] and cut designs out of sturdier material [5].

A similar project to the aforementioned do-it-yourself project is the Aikon project [1]. This involves a robot system which is able to sketch out images, namely of people's faces. The team's objective is to simulate the process of face sketching, and looks into how different artists begin drawing a face. Since the Table Stitch machine aims to take a simple sketch (beginning with simple shapes), we are also looking at the way machines interpret pictures. Like the issue with the DIY text-drawing project, we aren't dealing with the pivoting of a pen in our Table Stitch project--we're dealing with fabric and calculating the pivots and movements to stitch shapes. However, although Aikon's input is a photo of an actual person and ours is a sketch, we both want an output that is sketch-like--theirs in ink, this project in stitches, both having to be machine-interpreted.

Unlike the do-it-yourself projects mentioned above, one project manages to write by moving the table in addition to shifting the instrument itself [7]. This is ultimately closer to our project; as stated many times before, we cannot move the sewing machine itself. Watching this particular project in action, we can get a better sense of how feasible it is for the surface to move. With all the numerous projects already done with drawing some sort of pattern, ie a square [4], it is also possible that the same functions these projects use to move the pen can be used to move the fabric (so, switching around or looking at these already-created projects backwards).

The Egg-Bot [9] project also tries to accomplish a similar task. Rather than stitching, the Egg-Bot draws user input with a Sharpie on an egg. Similarly, what the Table Stitch is trying to accomplish can be seen as a 2D CNC tool. With the Egg-Bot, users draw a design in a program called Inkscape and the design is converted into G-Code. G-Code is a universal stepper motor language. 3D printers also use it. By breaking down an input into steps, a design can be replicated onto another surface.

Overall Design Description

Structure

The design for the Table Stitch starts with a pre-threaded sewing machine. A frame is responsible for holding the fabric taut, so that it does not get wrinkled during the stitching process. This prototype uses binder clips to hold the felt down. The Arduino Fio is connected to a computer in order to get the user-selected design and calculate the frame movement required for the proper stitches to be made. The frame is made of MicroRAX, connected to a set of 3 layers of MicroRAX in order to move the fabric into the shapes declared by the user.

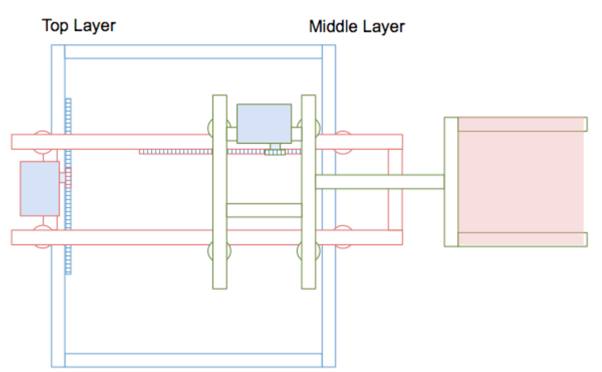


Figure 1: Rendering of complete system.

The machine consists of two moving layers, each layer made of 4 MicroRAX beams. These 3 layers are attached to four more vertical MicroRAX at each corner, connected to another bottom layer the same size as the first, in order to make a lifted base up to the height of the sewing machine. This complete structure fits into the sewing machine as seen in Figure 1. The red area is where the fabric is placed.

Each plate will move by means of a stepper motor held onto a MicroRAX beam by screws and an acrylic plate. A stepper motor provides more accurate movement across gear racks than a continuous servo would. The MicroRAX beams will be used to construct a series of three frames arranged in vertical layers. Here is an illustration of the top-most layer:

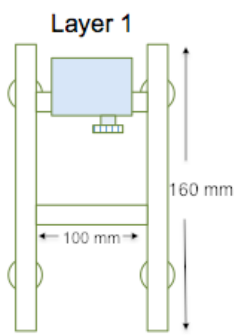


Figure 2: Rendering of top layer.

The frame is composed of two 100 mm MicroRAX beams connected to two 160 mm long beams. A stepper motor is screwed onto one of these 100 mm beams by means of an acrylic plate. This plate was designed in order to attach the stepper motor in a manner where its gear could reach the gear rack. We went through three iterations of this acrylic plate; the first design failed for two reasons. First of all, it placed the gear just barely too far from the gear rack. Second, its length proved just barely too wide for the MicroRAX beam. A second version was designed that fixed the second problem but still failed at the former. Finally, a solution was found in cutting slots where screws should go instead of just cutting a single hole; this way, corrections could be made to the height of the unit without having to cut another piece of acrylic.

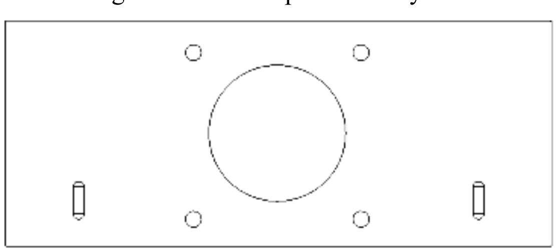


Figure 3: Acrylic attachment for stepper motors

On the bottom of the MicroRAX beams, there are rollers fastened with screws so that they may roll against the layer beneath itself. To be more precise, these rollers will touch the sides of the second layer, providing additional mobility so that the stepper motor may move the layer with as little resistance as possible. Of course, this gear will be used to traverse the gear rack of the layer below:

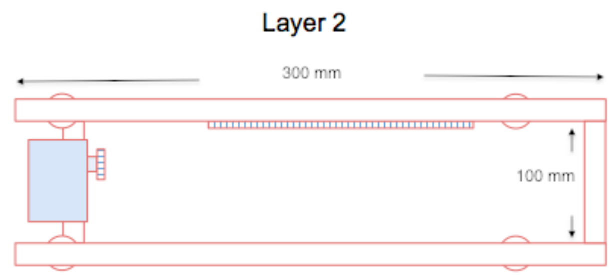


Figure 4: Rendering of middle layer.

The second layer is composed of two more 100 mm MicroRAX beams and two very long 300 mm MicroRAX beams. Attached to one of these 300 mm beams is the gear rack that links to the first layer's stepper motor. Much like the first layer, rollers and screws are placed beneath the four corners of the frame so that they may roll against the beams of the third layer. In addition, this layer is also equipped with a stepper motor that is also fitted with a Hitec gear. The top two layers will be attached to the bottom layer, pictured below.

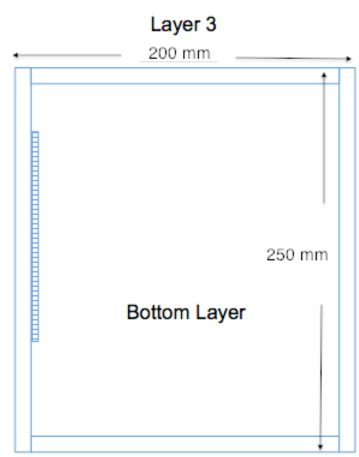


Figure 5: Rendering of bottom layer.

This layer's frame is formed by two 200 mm MicroRAX beams and two 250 mm MicroRAX beams. In addition, a few additional materials are necessary in order to join the MicroRAX beams themselves together. First of all, the beams of almost every layer will require two-hole nut plates fastened by 5mm button-head cap screws. This layer provides a gear rack for the previous layer's gear to move across.

Both stepper motors needed screws to fasten themselves to the acrylic plate seen in Figure 3, and additional screws to fasten them to the MicroRAX adaptors that fit into the MicroRAX beams on which they rest.

The horse-shoe shaped frame that holds the fabric consists of three 150 mm MicroRAX beams. In this case, we must employ joining "L" plates to fasten two of the parallel 150 mm beams to the other. This is necessary because of the fact we only have three beams to work with; nut plates by themselves would prove to be quite insufficient, as the components of the frames would not be nearly as rigid as they should be. Still, nut plates are good enough to at least bridge the fabric frame to the first layer by means of a 140 mm MicroRAX beam.

The frame is inserted between the operating parts of the sewing machine. The fabric is clamped taut in order to stitch upon it (for our first prototype, we attach paper with tape and use a highlighter to draw the shape that will be stitched. For our final prototype, we found that the machine we had worked best with felt, which we used—we used either black thread on white felt, or white thread on black felt). In addition, we go ahead with the assumption that the user will manually press the sewing machine's pedal. This has two benefits, the first one being the fact that we do not need to invest in any expensive devices nor spend the time fully calibrating them. And secondly, this will allow for users to cease operation of the machine instantly under extenuating circumstances. We also make the assumption that the user will hold the pedal down until the shape is completed.

Circuitry

The stepper motor coils and power cords were soldered onto the Big Easy driver (version 1.2). Along with the circuitry shown in Figure 6, pin M3 was set to ground for each Big Easy Driver. Setting M3 to LOW while M1 and M2 default to HIGH results is 8^{th} step movements.

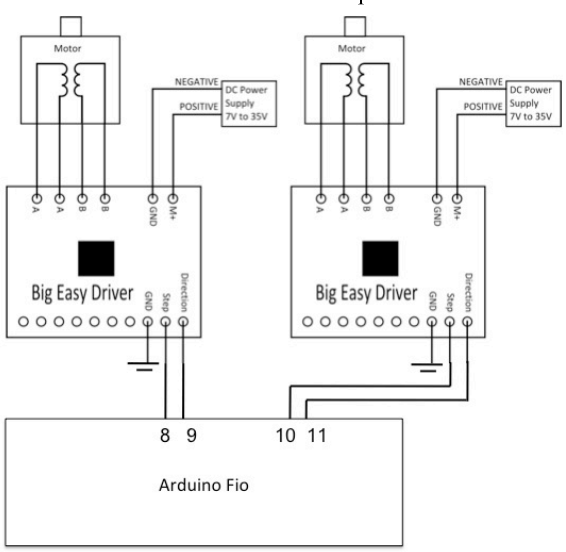


Figure 6: Schematic of circuitry. Image adapted from http://www.schmalzhaus.com/BigEasyDriver/

Our first prototype succeeded in moving the MicroRAX to draw a diagonal line, a square, and a triangle, seen in the video at http://youtu.be/obPWJ8LuZPs.

The first prototype was built without a sewing machine, so a sheet of paper was used to test the resulting designs. It was fastened to the MicroRAX by means of scotch tape. As you can see in the video, one person had to hold a highlighter still in order to mimic the sewing machine. There is separate Arduino code for the diagonal line, the square, and the triangle.

For the final prototype, we needed to make an elevated base for our Table Stitch to reach the sewing machine. First we tried to go about this by laser cutting a .25inch acrylic board:

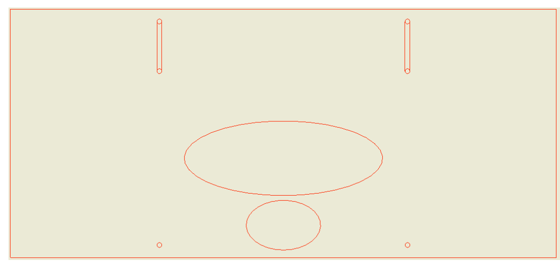


Figure 7: Acrylic base design.

We created a similar version of Layer 3 (Figure 5) out of MicroRAX beams for the bottom. Instead of using 250mm beams, 300mm beams were used. We believed that a larger base would be more stable than one of the same size as Layer 3. The small holes at the bottom of the acrylic would attach to the double of Layer 3, while the top holes would attach to the rest of the Table Stitch—those holes are longer to allow for height adjustment. However, when we tried to put it together, the screws would not hold up the weight through the .25inch acrylic. The solution was to use extra MicroRAX beams to form 4 short table legs and attach them with 90 degree join brackets. Given the nature of MicroRAX, this set-up allowed the height of the Table Stitch to adjust to the sewing machine.

On the next page, figure 8 shows the final assembly. This structure worked effectively for our final version of the prototype. Because the height is adjustable, Table Stitch will work effectively for different heights of sewing machines.

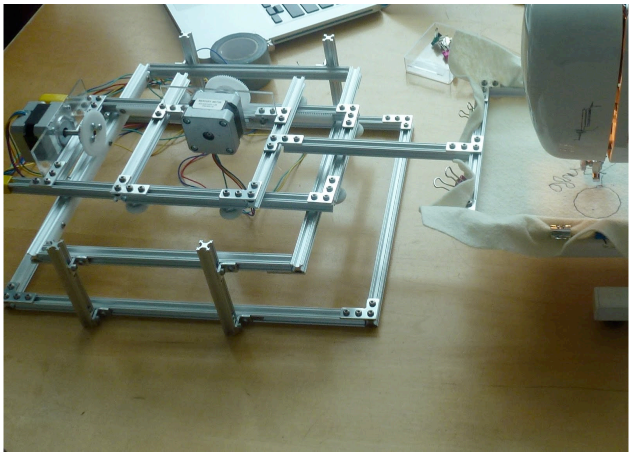


Figure 8: Completed system with MicroRAX base.

Programming

In order to allow these motors to draw shapes from coordinate data, an implementation of Bresenham's algorithm was necessary. This algorithm, which is normally used to convert coordinate data into a discrete form such as pixels, was retooled to work with stepper motor movements. This proved to be a highly natural conversion, as the code already worked by "walking" from point to point much like the table stitching machine does.

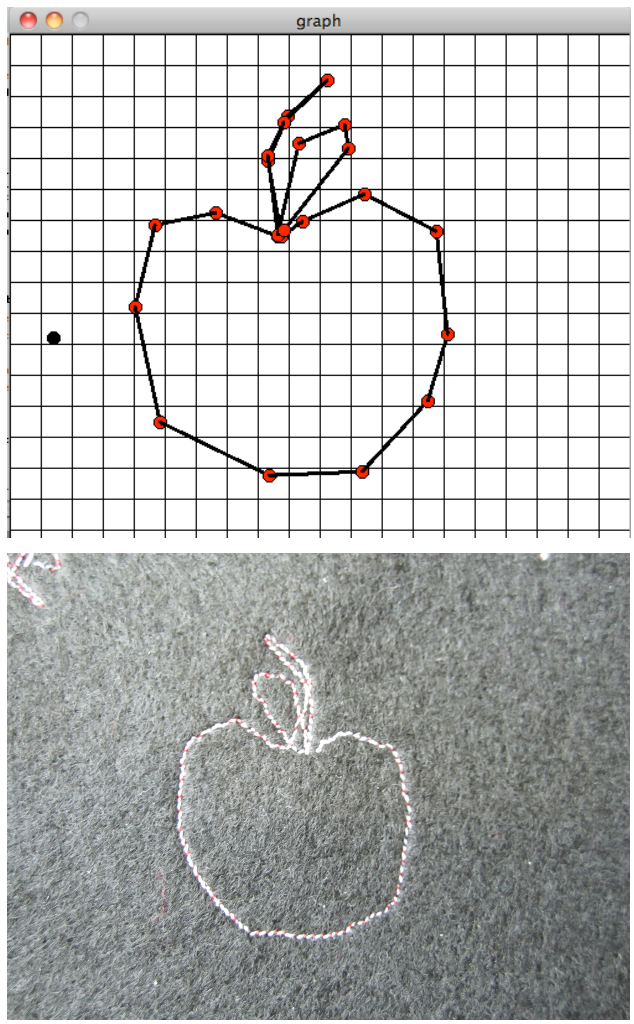
The algorithm works by moving along a line from point to point and deciding, based on the slope of the line, which direction (up, down, left, or right) it should move in order to match the original line best. When a line first starts, we find the difference between the current coordinates, (x0, y0), and the next ones, (x1, y1). We store a variable labeled "error" which is the difference of dx, the absolute value of the difference of x1 and x0, and dy, the absolute value of the difference of y1 and y0, times two.

This "error" variable is key in determining whether or not we will draw in a specific direction; if it is greater than negative dy and x1 is greater than x0, then we should draw right. If it turned out that x0 was actually greater than x1 we would end up turning left. In addition, if "error" is less than dx and y0 is less than y1, we move up; in contrast, if y0 was in fact not less than y1 we would end up moving down. This is how the algorithm principally decides which direction we should move at any given time.

The only thing left to take care of is moving our internal position closer to the current coordinates. Since our coordinate system exclusively uses integers, this is actually fairly simple. If we moved left, we would subtract 1 from our current position. If we moved up, we would add 1, and so on. While drawing a line we indefinitely move until our current point matches the next ones. At this point, we end the current loop and start from the beginning of the next line.

Of course, the only important thing left to consider in implementing are drawing system is regulating the speed at which the machine moves. Since we are sewing, not drawing, there are some very real physical limits as to how fast we can move. Consequently, delays are executed along with step motor movements; experimentation found that a delay of 20 ms worked best for both motors.

To add user input, a UI was created with Processing. When working with Processing, the Standard Firmata Library is uploaded to the Arduino and all of the code is written in Processing. Because Bresenham's algorithm functions on X and Y coordinates, user inputs from Processing can be fed into the algorithm. A window was created five times bigger than the estimated Table Stitch movement frame so that users could see the pattern they are designing. The actual coordinates used are divided by a scale of 5 which slightly distorts the stitched image.



Figures 9 and 10: User input and resulting stitching

Clicking within the window causes a red point to show up. On the next click, a line will be drawn between the previous point and the new point. Due to the nature of a sewing machine, this line helps to reinforce the idea that the image must consist of one continuous line.

If the user wants to clear the current points, they can press the delete key. If they would like to evoke the predefined patterns they press ‘s’ for star, ‘c’ for circle, and ‘l’ for “info” in cursive letters.” When the user is finished with their design, they can push any other key to begin stepper motor movements. A full demo of the user interface can be found at:

http://www.youtube.com/watch?v=OtETlalFXkU

The final demo with the second UI can be seen at:

http://youtu.be/mZwg7l5le8E

All Table Stitch YouTube videos can be seen at:

http://www.youtube.com/playlist?list=PLBCDBE046332CBE70

Future Work

In the future, we would hope to make the Table Stitch as accurate as possible. As it stands, the Table Stitch works accurately, stitching based on the desired coordinates plotted in the Processing interface. However, when the stepper motors overheats, the accuracy becomes a little wonky and steps occasionally slower, making the stitched work different from input.

Another issue with the current Table Stitch is that you can only stitch in a continuous line. We could have a beeper or a light that went on to signal the user to lift his or her foot from the pedal, but this produces problems—we don’t know if the user will be too slow, or if there should be a button pressed by the user to let the machine know it is ok to stitch again, and the timing will be complicated. Ideally we would spend money to be able to build an attachment to the sewing machine, or somehow alter the sewing machine’s wirings so that the user doesn’t have to press the sewing machine’s pedal at all. As it stands for the final prototype, we could not work directly with the wiring of the sewing machine.

Other future work could include having a larger area where the Table Stitch could stitch; having more colors to stitch with (another issue with working with a sewing machine); and make it easier to change fabrics. Incorporating Inkscape and G-Code would allow the Table Stitch to make advanced embroideries.

Lastly, the MicroRAX of our Table Stitch had to be held down for demonstrations because the stepper motors moved it too much. Although a good base was built to lift the fabric up to the level of the sewing machine, there was no way to attach this directly to the machine to prevent the stepper motors vibrating the fabric and skewing our stitched results—the Table Stitch had to be manually held down. Therefore, it might be a good idea to build the Table Stitch directly into a sewing table.

REFERENCES

1. Aikon. http://www.aikon-gold.com.
1. Brother Embroidery Machine Digitizing to Stitches. http://youtu.be/-eJ89lo7Hvg.
2. CNC Machine Writing with a Pen. http://tinyurl.com/autowrite.
3. Drawing machine progress. http://youtu.be/IxGHEMfeCHU.
4. Home made laser cutter 4 http://www.youtube.com/watch?v=FHNfl_26zyE.
5. How to do Free Hand Machine Embroidery. http://youtu.be/6U0_Eq7o0k4.
6. Mini CNC table made from computer parts. http://www.youtube.com/watch?v=Hlzs03bJD3E.
7. Self made CNC machine with DVD writer laser burns laser hazard logo on CD case. http://www.youtube.com/watch?v=sr5U5YlE9Zc.
8. Egg-Bot, http://egg-bot.com/
9. SMH How does a sewing machine work? http://youtu.be/II_bUNjmg78.