Sidewalk Chalk Printer

Andrew Eccles
Cornell University
Ithaca, NY
aoe3@cornell.edu

Catherine Schmidt
Cornell University
Ithaca, NY
cls352@cornell.edu

Claudia Gisemba
Cornell University
Ithaca, NY
ckg36@cornell.edu

Author Keywords
Chalk printing, Sidewalk chalking, rapid prototyping

ACM Classification Keywords
H5.m Information interfaces and presentation: Miscellaneous.

INTRODUCTION

We were motivated to create a sidewalk chalk printer because of our experiences with trying to advertise with chalk on campus. While this is an effective means of advertising an event, it is also quite painstaking to create, often involving hours of labor that require painful bending over and the added frustration of getting chalk on one's clothes. Thus, the idea of a chalk printing machine is a highly appealing one, and one that would solve all the problems currently involved with chalking while preserving its benefits. It moves a lot quicker than a human can and can spray multiple pixels of chalk at once, so it would cut down on time; its handles are reachable by a human standing upright, so there is no painful bending; and its downward facing nozzles get nowhere near the operator's clothing.

Overall, our design is a push-cart style machine, with two aluminum rod handles attached to a rectangular frame containing five spray chalk cans. The user can push the printer forward and backward thanks to three wheels attached to the frame. The cans' nozzles are triggered by servos at times which are dictated by the code, which is sent to them through an Arduino. The code contains functions for each letter and number, and many symbols. A user can input a word or phrase to be printed simply by calling these functions in sequence. The code then triggers servos based on a five-by-five grid of pixels for each letter, number, or symbol. Once each pixel for a specific column of the grid is called (for example, in the letter B, every can would be triggered in the first column to create the straight line forming the leftmost part of the B), the user can push the printer forward and wait for the next column (again, using B as an example, the next cans to be triggered would be the first, third, and fifth).

Our main design goal was to create a machine that could spray any words a user inputted into it. We also wanted it to be mobile and quick and require little labor from the user. At first, we wanted it to be automated and be able to move itself. However, after running into some issues with our motors we decided this wasn't feasible and decided to attach handles, which has the added benefits of allowing the user to control the spacing of their letters and push the cart in patterns other than a straight line.

RELATED WORK

There are different implementations of a chalk printer from which we draw our inspiration. The implementations vary in terms of purpose and extent of complexity. Most implementations print preset characters and design and are powered by an Arduino or being highly mechanized such as the Nike Chalkbot.

The ChalkJet[1] is "a semi-automated printing system which uses chalk spray to produce large scale text and images on the ground" (1). This implementation provides a solution to the challenge of advertising on sidewalks given that it "efficiently utilizes the ground space as a means for distributing information to a target public audience"(1). While this is crucial, it is important to note that the printer uses one color "with no shading or gradients"(3). This limits the power of the text and images produced as one color may be perceived as boring.

The Nike Chalkbot[2] was an "automated printing system used print messages on country roads to raise awareness for a cancer foundation. It was used as an advertising tool which integrated hardware, software" (2) and the public. The Chalkbot used water soluble chalk-based paint made from soy protein. The choice of paint was influenced by the purpose of the Chalkbot given that the paint was non-cancer causing, hence environmentally friendly. The robot is driven similarly to a normal vehicle. It is a “hydraulic robot run by a laptop” (2) and the images/text are preloaded before printing. The preloading is done using Wi-Fi connection. The image/text is then printed on the road using chalk spray used in line painting

The Paint Machine[3] creates “simple bitmaps with spray chalk on street sidewalks”(3). The machine is Arduino controlled and is mainly powered using an 11.1 V battery. The main power source was faulty in that it “got extremely hot”**.** This coupled with the Wi-Fi connection constantly breaking presented a challenge in rendering a consistent image.

The Txtbomber is an “Arduino powered high-tech graffiti printer which uses seven built-in pens”[4]. It prints letters. Since it is Arduino powered, there is no need to use a computer.

Our project drew inspiration mostly from the ChalkJet, Paint Machine and Txtbomber. We therefore designed an Arduino-powered chalk printer which uses a 12V rechargeable battery to power the machine given that we needed to power 2 stepper motors and 7 servos. We were aware of the potential overheating of the electrical components as experienced in the Paint Machine. Consequently, we strategized to hook up one component at a time while investigating possible causes of overheating. This would facilitate easy debugging of problems in the circuit.

We then chose to directly upload the program onto the Arduino microcontroller as opposed to using a Wi-Fi connection to send information. This would eliminate problems potentially caused due to poor internet connections, which happened to the Paint Machine, and which we anticipated encountering at the Sciencenter.

We also took influence from the ChalkJet in its use of multiple cans as opposed to a few cans that move back and forth on a pulley. Our initial design was the latter, but after running into our own problems, and talking to teaching staff, we decided that a seven-can approach would be best. The ChalkJet uses eight cans, but given that we could draw out a grid of every symbol we needed using seven cans, we decided we did not need to go that far.

We took further inspiration from the ChalkJet in that it is also a push-cart style machine. It has a roughly three-feet high handle for a user to grip and push, and that is how it is moved, as opposed to the automatic motion we originally tried to implement.

OVERALL DESIGN

Video of Printer Working

https://drive.google.com/open?id=0BzFKhfb5YEPCNElNUnhhY0FPc0k

Code

The code was first implemented as intended to be used by a user typing in whatever they wanted to be printed into the serial port and the code reading the input and sending the information to the printer. This was done by creating a large list of if-statements, one for each letter, number, and symbol we would allow to be printed, that dictated what each of seven servos would do for each column of a five-column grid (the seven rows by five columns grid being how we designed each symbol). For example, for the letter B, for the first column all seven servos would be activated (moved to their “active” degree), and then the next column only the first, fourth, and seventh servos would be activated and the others would all move back to their “resting” degree. Between each column the code made the stepper motors move forward one rotation to be able to print the next part of the symbol.

This first code was long and clunky and the serial input was hard to do. There were also some issues with how the servos were being called that caused them to interfere with each other. Therefore, we streamlined the code a bit by getting rid of the if-statements and replacing them with an individual function for each letter, number, and symbol (e.g. printA()). The code still was relatively the same within these functions, but we discovered we needed to call the servos one at a time to prevent them from interfering with one another. Also, we discovered that each servo began at a different angle, and thus the angles in the code were dependent on our testing how each servo moved before we began. Also, once we got rid of the motors, we got rid of the code for them as well.

In the end, the code functioned well. It was easy to input a word or phrase to be printed, as all you had to do was type in the functions you wanted in order. Once we calibrated the servo angles, they moved how they were supposed to, when they were supposed to.

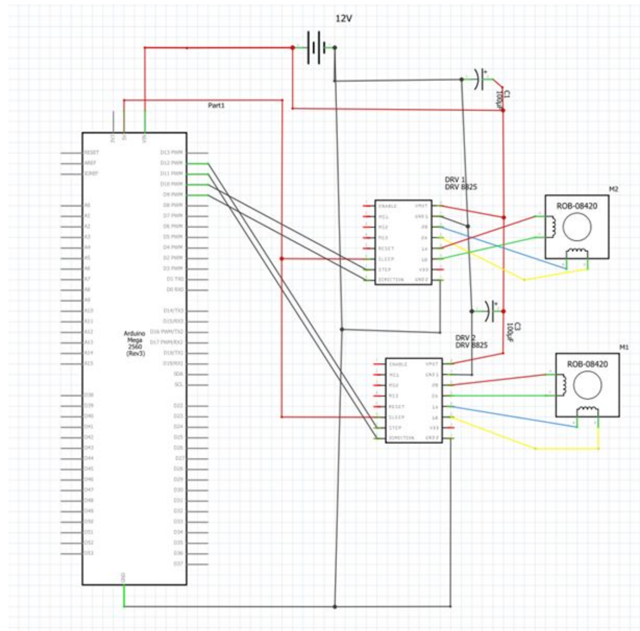
Circuitry

The components include:

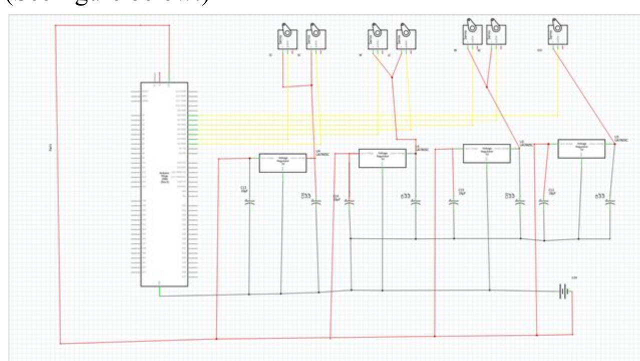
1 Arduino Mega 2560 microcontroller
2 Stepper Motors
2 DRV8825 Stepper motor controllers
7 5V Servos
4 UA7805C 5V Regulators
2 100µF capacitors
4 10µF capacitors
4 1µF capacitors

The circuitry was first implemented using a Photon as the microcontroller. In this phase, we were only able to power and control the 7 servos and one of the stepper motors using a 9V wall outlet power source. In as much as every component worked seamlessly, we realized the ineffectiveness of this approach given that one of the goals was to design a motorized sidewalk chalk printer. Consequently, we switched to the Arduino Mega 2560 microcontroller which provides multiple PWM pins and facilitates direct connection to a larger power source. Initially, we anticipated to use 3.7V Lithium-ion battery but realized we required a larger power source to power 2 stepper motors and 7 servos. We opted to use a 12V Lithium ion battery as a larger power source. Secondly, we powered the servos using a the 5V connection from the microcontroller which constantly made it heat up.

We finally used two separate breadboards to complete the circuit: one for the stepper motors and the second for the servos. On the breadboard for stepper motors: 12V power is directly connected onto the breadboard then to the the VIN of the microcontroller. Each stepper motor is connected to a DRV8825 motor controller then to the microcontroller using the driver and step pins from the controller. (See figure below:)

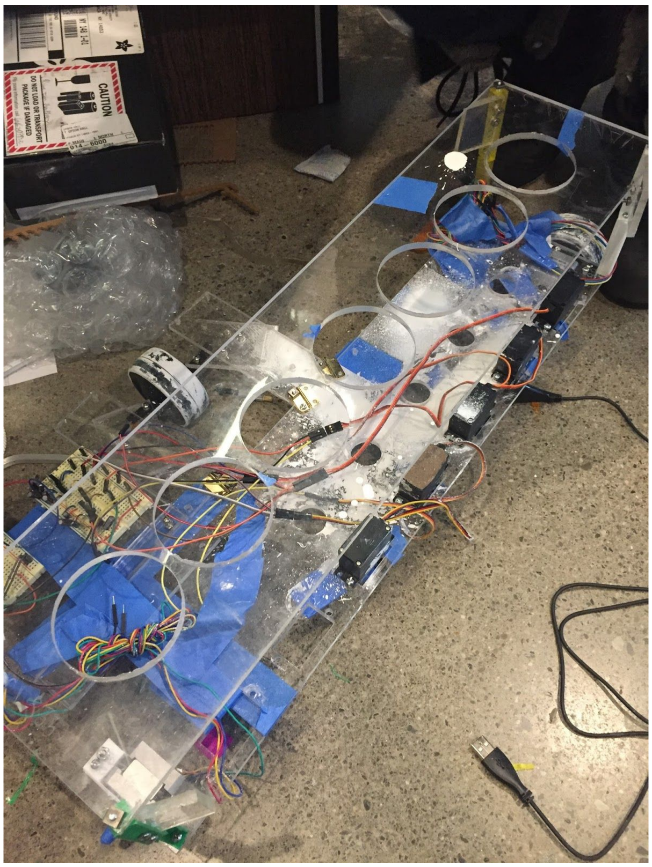


On the second breadboard: we use UA7805 regulators to convert 12V to 5V which power the servos. Each of the servo is connected to the PWM pins of the controller. For all this to work seamless, we upload code onto the microcontroller which facilitates the printing characters. (See figure below:)



Frame

The frame is made of acrylic sheets to make an open rectangular box-like structure which measures about 2 feet by 1.5 feet. The structure consists of two sheets of acrylic interconnected with 3D-printed linkages layered to form the structure. Each sheet is about 2 feet long by 3/8 inch wide. The structure is built to hold 7 chalk spray cans. To hold them in place, the first sheet from the ground has 7 circular openings wide enough for the spray can nozzle. It also has seven servos screwed into openings, each one positioned a few inches from the nozzle hole. The second from the ground has circular openings that measure the diameter of a chalk spray can. This allows the can to be easily slipped in place. There is also attached to the lower sheet two smaller, rectangular sheets which jut out the front and hold one wheel on an axle. There are also attached on either side of the lower sheet two 3D-printed holders that hold stepper motors which are connected to a wheel.



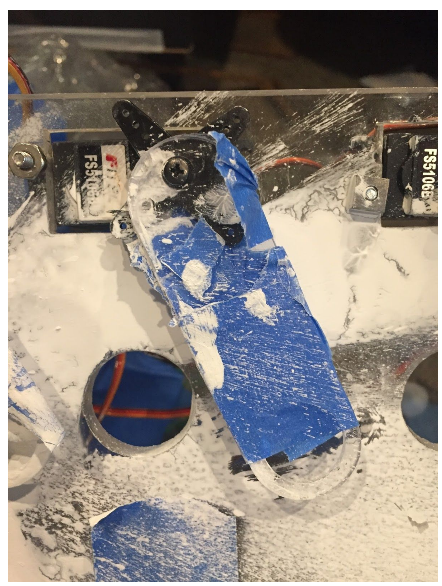
The frame.

This feature was first implemented using cardboard to make a rectangular box of about 2 feet by 1.5 feet. On the bottom, one of the longer sides had a slit in it - this was the "vent" through which chalk could escape. On the outside of both sides of the frame, in the center bottom, were cardboard wheels, roughly two inches in diameter, held in place by thumb tacks. Inside the frame, a thin strip of cardboard spanned the entire width of the box. This represented the slider that we thought would move the spray cans. Attached to this strip was a cardboard "pocket" holding four batteries which represented four spray cans.

Later prototypes were very similar to this initial one. While their execution was more expert, the design remained the same. We were very dedicated to the concept of a slider with a holder for cans in our initial conceptions of this project. However, once we started building and thinking about how actually to execute this, we ran into a few issues: how to hold the cans up, how to trigger them, how to move the slider, etc. This led us to our current and final design of using multiple cans instead of moving cans and using cans that point straight down, making unnecessary any vent or funnel, as we had initially planned upon.

Spraying

The spray cans are in fixed position. They are placed in position through the circular openings on the frame. The spray cans are triggered by a servo-connected acrylic lever pushing on their nozzle. The servo is attached to the frame layer closest to the ground. Each spray can has a corresponding servo – 7 chalk spray cans and 7 servos. The servo triggers the nozzle as it sweeps at an angle determined by where each individual servo starts. This is implemented using laser cut levers screwed onto the servo attachments such that each time the servo sweeps an angle, the lever moves and triggers the nozzle to spray chalk. The levers are about 2 inch long by 1 inch wide.

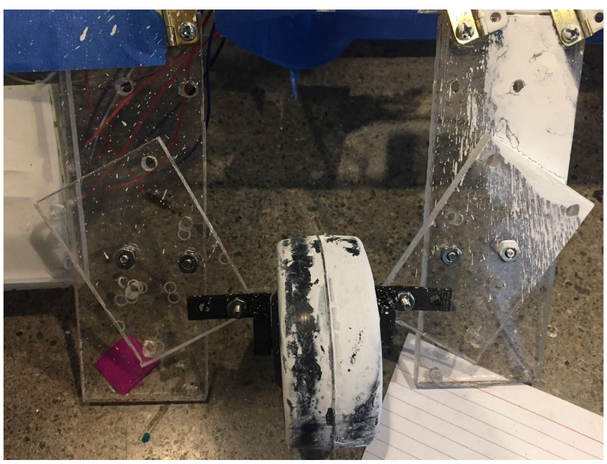


A lever attached to a servo.

Initially, we used wires to trigger the nozzle as the servo made a sweep of 130 degrees. In as much as it worked some of the time, the wires often came off the servo attachment making it impossible to trigger the nozzle. With this in mind, we modified the triggers to a 3D-printed raindrop-shaped part that was also attached to the servo attachment. When tested, the part was not strong enough to trigger the nozzle. It would work some of the time leading the the release of too much or too little chalk spray when the nozzle is triggered. This was ineffective. Consequently, we chose acrylic levers due to their sturdiness and strength. When implemented, the acrylic levers applied adequate pressure on the nozzle to release a good amount of chalk spray.

Movement

In our last prototype (which was cardboard) we implemented movement using three wheels. The first pair of wheels were placed at one end of the frame being controlled by a motor. There was a third wheel at the middle of the frame to provide stability such that the printer did not drag or scrape. All this was accomplished while using the power supply of plugging in the Photon to the computer.



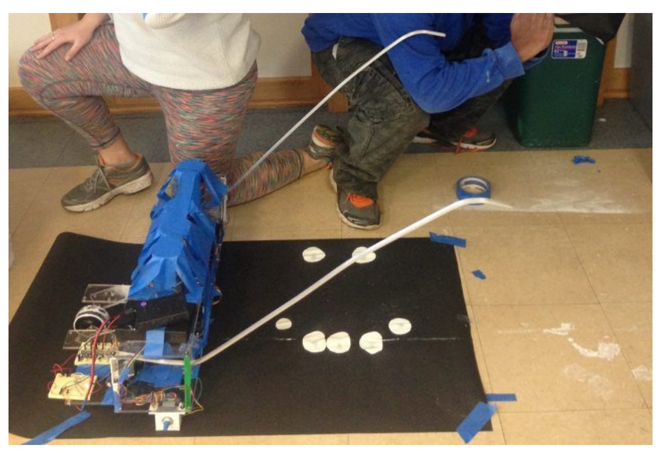
The third wheel.

For our final design, initially we implemented movement using two pairs of wheels held in place about 2 inches from the frame using pieces of acrylic about 5 by 4 inches attached to the frame. The wheels were mounted onto the frame using 3D-printed parts and stepper motors used to move the wheels forward and backward. To evaluate performance, we assembled the frame and tested movement. We found that the frame would not be stable enough to move and hold 7 cans at same time.

We modified the design to have one pair of wheels centered at the frame and a third wheel at the front for stability. The third wheel is held about 5 inches from the frame. Each of the two original wheels has a stepper motor used to move both wheels simultaneously. The stepper motors are held in place using 3D-printed holders.

During integration, the main problem we faced was that one of the wheels would slightly bend during movement. This hindered seamlessly straight movement. This was attributed to loose attachment of the stepper motor to the wheel due to the smoothness of the surface. Additionally, the stepper motor would jerk in the 3D-printed holders. To solve this, we modeled a tighter fit for the stepper motor mounts and used tape to create a grip between the wheel and the stepper motor. As a result, the wheel movement became seamless and both wheels did not bend during movement. However, the weight of all the cans was too great for the motors to be able to move the frame forward; it was only once we took out all but three cans that the printer could move on its own, and even then it was very slow.

Therefore, we decided to replace automated movement with a push-cart style system. We bent two long aluminum rods (roughly four feet long each) on both of their ends and laser cut two acrylic rectangles, roughly 2 by 3 inches, with rectangular slots in each that were the width of the rods. We then attached these rectangles to the 3D-printed frame linkages. The ends of the rods were inserted into the slots and thus we had a sturdy means of pushing and steering the printer. The wheels remained connected to the stepper motors but we no longer had the code call to the motors at all and merely used them as an axle for the wheels to move on.



The entire design, with the aluminum handles.

FUTURE WORK

Going forward, if we were to continue to work on this project, we would probably want to do a few things differently. First of all, we would get new servos, as one main obstacle we kept running into was that our servos would work erratically or not at all. This was probably due to a variety of factors, including wiring and spray chalk getting into their mechanisms. Therefore we would also look into how we power and wire all the servos, and make sure not to let the chalk actually near the mechanisms.

We also would maybe see about making the movement actually be automated. As it is, we like the push-cart method because it allows the user more flexibility and control over their design, and it relies less on the motors working which is not always guaranteed. However, if we were to make it automated, we would probably want a separate power source for the motors. Having one power source go to both the motors and all the servos weakened the input to all. Additionally, we would probably also implement another motor on the third wheel which currently does not have a motor. This would provide more overall power to movement. We would also work on creating a more sustainable system of gripping the axle to the wheels than tape.

Another thing we might implement in the future would be a way to input a word over a website, instead of just modifying the code. Calling functions for symbols was a fine and easy way to make the printer work, but it would be more elegant and usable to enable web form submission. At the Sciencenter it was just us inputting things to print, so it was okay, but if we wanted other people to be able to input their own words then a more global interface would be beneficial.

On the whole, there was a lot that we learned when building and testing the final product. Much of what we planned on functioning correctly didn't and we had to make many last-minute changes to accommodate the real-world limitations we ran into. Overall, if we went forward with the printer, we would work on polishing up the solutions we implemented and making our final design, which we are more or less satisfied with, more elegant and usable.

ACKNOWLEDGEMENTS

We would like to thank the following people, who were instrumental in our completing this project:

- Francois Guimbretiere
- Roshini Thiagarajan
- The teaching staff of INFO 4320
- Tone Xu, Tiffany Zheng, and Cyndi Chin for offering their servos last minute

WORKS CITED

[1] Cook, Jeremy. The ChalkJET writing machine. http://makezine.com/2014/08/25/the-chalkjet-writing-machine/

[2] Alvarez, Marcelino. The Nike Chalkbot – Five years later. https://uncorkedstudios.com/blog/the-nike-chalkbot-five-years-later

[3] . Paint Machine. https://tinkerlog.com/2015/11/17/paint-machine/

[4] Szcyzs, Mike. Txtbomber. https://hackaday.com/2010/08/05/txtbomber/