Thesis Project Portfolio

Signature Replication Machine

(Technical Report)

IS SAAS THE FUTURE? A CASE STUDY OF SHOPIFY'S BUSINESS MODEL AND SOCIAL CONSTRUCTION

(STS Research Paper)

An Undergraduate Thesis

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Zichao Hu

Spring, 2022

Department of Computer Engineering

Table of Contents

Sociotechnical Synthesis

This portfolio contains two major pieces of work completed throughout the school year of 2021-2022. The first piece is a technical report about a capstone project that develops a signature replication machine by a group of three fourth-year computer engineering major students. The second piece is an STS research paper that investigates a popular state-of-the-art model called software-as-a-service (SaaS) by performing a case study on Shopify Inc.

In the capstone project, a key motivation is that signing documents can be very time-consuming and repetitive but it is also prevalent and crucial in everyday life. It is desirable if we could let machines take care of such repetitive tasks for us in daily life. As a result, the goal of this capstone project is to design a fully functional signature replication machine that frees humans from signing their names physically and repeatedly. The main UX design of the project is that the users only need to provide a digital signature via writing on a touchable pad and then the signature will be sent through the Bluetooth network that enables the machine to replicate the signatures exactly as specified. The members of the team collaborate seamlessly to carry out the project. Finally, the project is demonstrated to and tested by various groups of visitors on the showcase day, and the machine can reliably and successfully replicate any signature provided by the users.

In the meanwhile, the STS research paper looks into a product of the ongoing digitization efforts today, which is the software-as-a-service (SaaS) model. The SaaS model is revolutionary in many aspects of society today. It both influences and is influenced by different groups of people, such as the company that adopts the SaaS model, the customers of that company, developers who contribute to the SaaS platform, and the investors who invest in the company's stocks. Specifically, the work will focus on a SaaS company called Shopify, and the work will analyze Shopify's business model to gain a deeper insight into the SaaS model. In addition, this work will employ the social construction of technology (SCOT) as the primary methodology for investigation. The work studies how different groups interpret the SaaS model and understand the role of SaaS in society. Furthermore, the paper finds interpretations that give rise to conflicts of interests and provides an analysis of how each conflict could be resolved through closure.

The capstone project is a typical IoT project that aims to use modern technology to simplify traditional tasks. This aligns with the focus of the STS research paper that explores how SaaS, a modern technology, exists to address traditional problems today. The STS paper takes a step further to study the technology in a social context via the social construction of technology methodology. This methodology argues that technology does not determine human action but instead, human action shapes technology, which in turn could contribute to the understanding of the capstone project in a social context as well.

Signature Replication Machine

A Technical Report submitted to the Department of Computer Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Spring, 2022

Technical Project Team Members

Zichao Hu

Edward Lee

Yu-Jiyun Tao

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature _Zichao Hu_______________________________________ Date 05/03/2022

Approved ____ Date __________

Signature Replication Machine

Zichao Hu, Edward Lee, Yu-Jiyun Tao

12/17/2021

Capstone Design ECE 4440 / ECE4991

Signatures

Zichao Hu

Edward Lee

Yu-Jiyun Tao

Statement of work:

Edward Lee

Edward worked extensively on the image processing algorithm that interprets an input image file and converts it to coordinates usable by the signature replication machine. Edward was also responsible for taking the output of the image processing and transferring it to the microcontroller via Bluetooth. He was the primary embedded systems programmer for this project as well. He was responsible for taking in the command inputs and controlling both motors and the solenoid to produce the expected result. He also integrated all of the subsystems together into making the entire signature replication work as a whole.

Yu-Jiyun Tao

Yu-Jiyun Tao focused mainly on the mechanical process of the project. He was responsible for building the physical framework, designing the load, and managing the wiring of the system. Yu-Jiyun was also responsible for the main mechanical parts selection such as motors, switches, and aluminum extrusions. His subset of responsibilities included ultiboard design, integration testing between the programing and the embedded system, and soldering electrical components to the pcb.

Zichao Hu

Zichao Hu was mainly responsible for the PCB design and the motor selections. He inspected the datasheets of various electrical components and put in the purchase order. He designed the schematic and created the ultiboard layouts. Zichao also was responsible for sending and retrieving the board and the parts to 3W for assembly. His subset of responsibilities included creating a python app on the server sider to record the user’s signature input, which was subsequently used on the demonstration day.

Abstract

Signature is prevalent in everyday life and crucial for the validation of legal documents. However, it can be repetitive, especially if a job requires signing large amounts of legal documents. This project aims to free humans from signing their names physically by building an auto signature machine. This machine is created using a chassis that contains stepper motors that will replicate the motions of the signature and a microprocessor to receive the commands and translate them into electric pulses [1]. The MSP432E401Y is chosen as the microprocessor due to its computing capability and built-in supports for Bluetooth and security features. To use it, the user will take a photo of the signature and send it to our main program on the laptop. The laptop will wirelessly connect with the microcontroller, and the microcontroller will interpret the data and convert it into signals to motors to write down the signature. As a final product, the machine was not only able to replicate signatures, but also drawings and handwriting samples as well. The end product resulted in a machine that surpassed its original intent and can be used for any form of replication involving handwritten or hand-drawn samples.

Background

Signatures have always been a source of security and privacy for past centuries. The State of Frauds Act, which was passed in 1677, developed the concept of using signatures to seal contracts and acknowledge the approval of given documents [2]. Even in today's status quo, where technology is constantly replacing many traditional means of security, signatures are still widely used as a source of biometric protection [3]. Signatures can be seen using the most on the upper-level management of companies and government officials. It was recorded that Harry Truman had individually signed over 600 signatures each day during a brief period after the end of WWII [4]. Implementing an automatic signature machine would boost the efficiency and focus of the users and mitigate the possibility of injuries such as RSI (Repetitive strain injury). Furthermore, the machine can serve to sign papers for users who are not able to physically.

This machine is different from past products, such as the Ghostwriter autopen series, which requires the signature to be formatted and preinstalled in a customized SD card/USB [5]. Instead, this machine allows more accessibility by only requiring a picture of the signature. Unlike the previous products, it also can be activated wirelessly through Bluetooth implementation. Also, this machine can be used for any generic drawing or handwriting sample which supersedes the previously mentioned products.

Constraints

Design Constraints

The capstone course itself placed a constraint on the project. This project had to include a microcontroller or some form of embedded system and secondly had to include a custom printed circuit board that the team designed itself.

There were a significant number of design constraints due to the nature of the project. The entire system must be both large enough to encase a piece of paper for signature replication and small enough to be usable by users. The motors have a limited range of frequencies that were within the specifications of the datasheet [6], this would mean that lines above or below a certain slope value were unable to be drawn. The solenoid had to be strong enough to lift a pen and also provide enough pressure for the pen as well.

Logistical constraints include time as a PCB took around a week to develop and another couple of days to populate which puts a time constraint on the project. Part availability was another major constraint as the parts on the initial PCB design could not be reused as those parts were unavailable by the time the second PCB was designed.

There were also CPU limitations as the image processing was significantly too intensive on the microprocessor which required a server to process the image and send coordinates to the microcontroller. There was also another design constraint which was that multiple booster packs were required as one was required for the Bluetooth server but another custom booster pack was required to use the stepper motor and solenoid drivers using logic inputs from the microcontroller.

Economic and Cost Constraints

The budget for this project was five hundred dollars per team. This meant that sum of the costs of all materials that were purchased had to be below five hundred dollars. The team was able to save some money by reusing some materials that pre-existed in the lab prior to the start of the project but had to purchase most things such as the motors, printed circuit board, and its components.

Environmental Impact

Our project was mechanically built from aluminum rods and PLA plastic which are both very eco-friendly. Aluminum can be recycled and PLA is biodegradable [7]. However, the microcontroller and PCB are not as environmentally friendly. The materials required to build the microcontroller and PCB are not as environmentally friendly, but these effects are unavoidable due to the constraints placed on the project itself. PCBs should be recycled to avoid adding harmful toxins to the environment.

Our project also has a light energy impact. If the microcontroller is powered, it will constantly be ready for a Bluetooth connection and draw a slight amount of energy constantly. While this is mitigated by the use of the low-energy mode that is naturally built into the firmware in the TI RTOS of the microcontroller [1].

Sustainability

Currently, this machine is not able to be used for long periods of time. The motors and electronics get very hot after consistent use and there were multiple times where there were slight deviations from the intended image that occurred as a result of the electronics getting too hot and did not output the desired result. Also, the base that holds the paper in which the signature is replicated is not as sturdy as desired and after long periods of use, the base can drift lower causing the pen tip to not touch the paper anymore. This would result in no signature replication and this can be changed by developing a much sturdier base so that the force of the solenoid and pen do not move the base over time. Future designs would include higher-rated electronics that would not overheat after extended use and also a sturdier base that would negate the two issues above completely. There are also concerns regarding each individual component used in this device. it is estimated that the MSP-series microprocessor and PCB can last up to 20 and 60 years, respectively, the stepper motors and sensor have a much shorter life span of fewer than five years. [9][10][11] However, as mentioned under manufacturability, all mechanical components that will be implemented can be easily replaced, making this machine quite sustainable. Furthermore, the machine's longevity can be improved with the right maintenance, such as lubrication. As long as there is a consistent standard voltage of 120V with a standard frequency of 60Hz (US standards based on International Electrotechnical Committee), this system can potentially run up to 5 years without adjustments and over decades with replacements [12].

Health and Safety

There are a couple of health and safety concerns regarding the machine during replication. This machine involves moving components and could cause damage to anyone who interferes with the machine. There are emergency switches that are built in just in case the motor movement would cause the current position of the pen to hit the outer edges of the devices. These switches act as an emergency stop functionality when pressed and prevent the device from destroying itself when it hits the outer edge. Regarding the electronics, there were a lot of concerns during development and the team tried their best to limit the number of exposed wires and electronics; however, the electronics should be handled with much care. A user should limit the interaction with the printed circuit board and electronics along with preventing any body part from interacting with the machine at all when the machine is moving. These two steps of caution should always be taken when using this device. Any information regarding information safety will be discussed during the privacy and ethics section.

External Standards

The project used the following external standards.

1. *Barr C Coding Standard* - The Barr standard is a set of rules for code that is adopted by a team of programmers to reduce the number of bugs in firmware by practicing and adhering to a certain set of rules. All of the embedded microcontroller code adhered to this standard [13].

2. *NEMA 17* - This was the standard for the stepper motor that was chosen for our project. NEMA 17 means that the stepper motor has a 1.7x1.7 inch faceplate [14].

3. *IPC 2221A* - IPC 221A is the industry standard for PCB designs. This standard specifically sets the part and track spacings of the board designs. This also sets acceptance criteria for PCBs including edges, material, holes, solder mask, and much more [15].
4. *IEEE 802.15.1-2002* - This is the IEEE standard for telecommunications and information exchange between systems regarding wireless data transmission. Since our project is using Bluetooth Low Energy devices, the Bluetooth transmission programs on both the server-side and microcontroller adhered to this standard [16].
5. *PEP 8* - This is a coding standard for Python programs to also increase the readability of the program and reduce the number of bugs. The server-side application adhered to this standard to increase the scalability and readability of this code while minimizing the potential for risk [17].
6. *IPC 7351* - A couple of surface mount components were used in the design of our PCB. This standard defines the size of the 0805 rectangular passive components that simplify PCB designs [18].
7. *NEMA 5-15* - NEMA also defines standards for the power supply plugs and the power supply that was chosen for the design followed the NEMA 5-15 standard which states that it can take up to 125V and 15A [14].
8. *STL (Standard Tessellation Language)* - The STL standard is a file format for storing and transferring information about 3D models [19]. The format was used to communicate geometrical surface information between the computer and 3D printer hardware.
9. *USB (Universal Serial Bus)* - USB is a standard that defines communication protocols between computers and devices [20]. The USB interface had multiple applications for the project. USB communications were used to connect laptops to embedded systems during the development stages of the project. This was essential in monitoring the behavior of the embedded systems as well as deploying developed code to the microcontrollers.
10. *TO package sizes* - This standard describes the PCB footprint for transistors [21]. This project used a BJT to control the solenoid, which uses the TO-220 standard.
11. *AES encryption standard* - This project required encryption to maintain the safety of the user's data. The AES standard is the industry standard for encryption and is a publicly accessible cipher that defines an algorithm to be used for the encryption process [22].

Tools Employed

Mechanical

Autodesk Inventor was heavily used to create CAD (computer-aided design) [23]. Not only did using CAD enable visualization of concept designs, but it was also utilized in printing 3D components. A band saw was also very useful in cutting accurate measurements for the aluminum extrusions. In terms of micro-measurements, having a dial caliper proved to be very useful in optimizing the height of the base and the symmetry of the V-slot framework. Allen keys and hex screwdrivers allowed me to assemble and modify the framework of this project. Furthermore, a scale was used to measure the mass of the main load. In addition, the 3D printer “Makerbot” [24] served as a critical component to print solenoid-load designs and microprocessor mounts.

Electrical

Multisim [25] and Ultiboard [26] were the major two tools used for electrical designs. Multisim was used to create schematics of the circuit while the Ultiboard was used to build the physical layout of the PCB board. Advanced Circuits [27] was used to order the PCB with the traces from Ultiboard and its FreeDFM tool was a free website from Advanced Circuits used in the project to verify the correctness of the PCB layout. For testing purposes, the NI VirtualBench [28] was used to debug the hardware in our project to check voltages at certain test points and apply voltages using a function generator. In terms of wiring and preliminary prototyping, solder iron was a necessary component to ensure strong secure connections.

Firmware

This project also used Code Composer Studio [29] which allowed embedded programming in the C language along with the TI RTOS which was the operating system that the microcontroller ran on.

Software

Python 3.8 [30] was used along with PyCharm [31] as an integrated development environment. The algorithms for computer visualization and image processing were done using the OpenCV [32] module and the Numpy module [33]. The Bluetooth transmission used another library called Bleak [34] to transmit to Bluetooth Low Energy devices.

Summary

This project allowed the team much more in-depth knowledge of using CAD to design mechanical structures, using Multisim and Ultiboard to design electrical schematics and TI RTOS for embedded systems.

Ethical, Social, and Economic Concerns

Social and Economic Concerns

This project’s intended audience is people who have a lot of signatures that they would have to sign. Instead of signing multiple times on a myriad of different pieces of paper, they could sign once and have the machine replicate their signature for the rest of the documents. This would provide a new alternative to replicating every signature by hand. Also, this project could affect people with disabilities or injuries such as arthritis as they would be able to use images of previous signatures to sign current documents. If this device is widely used, this could affect society by giving them a viable alternative to signing a large number of documents by hand. Widespread use of this machine could also change the legality and validity of signatures that were written machines. Since this device can also perform handwriting samples and drawing as well, the users would be able to replicate any form of writing they would like could allow physical representations of virtual drawings or writing.

Ethical Concerns

One major ethical concern in our project is about personal privacy and security. Since a person's signature is his/her identity in our society, having the ability to replicate one's signature exactly could cause privacy breaches. This could lead to disastrous consequences for individuals, as the replicated signatures could be used for fraudulent purposes. Although the machine has built-in security defenses such as AES encryption [22], there are still chances that a malicious person could steal another person's identity via other loopholes, such as hacking into the person's computer and subsequently gaining access to the signature machine. Unfortunately, this type of loophole cannot be defended against because the app of the machine runs on a host machine and if the host is compromised, the malicious attacker could easily gain backdoor access to the app and thus compromise the app as well. In addition, this same problem applies to many other appliances today, ranging from smart home devices to home cameras, printers, etc. Therefore, if we would create this product commercially, we would put an easily distinguishable label on our product warning the users to be extra careful with the security of the computer host. Hopefully, this course of action would be effective to defend against malicious attackers.

Intellectual Property Issues

This project has a decent potential for being patented as an invention for the same purpose does exist, but this invention performs the purpose of replicating signatures in a different method compared to this project. However, the idea of replicating and printing out signatures would definitely not be patentable. The first patent that is similar is a completely electronic device that can replicate handwriting samples and copy them in the software that it is currently running in, similar to a copy and paste feature in the software [35]. This is fundamentally different from our project while it may have a similar algorithm for capturing the strokes from the user's input in our touchscreen application, there is no physical replication of the signature on paper. The second patent is a method to capture a handwritten signature electronically, so it is doing the reverse of what our project is [36]. Our machine is converting an electronic version of a handwritten signature to a physical replication of the signature. The last patent is the one most related to our project. This patent is for the original autopen itself that physically replicates a signature and was in use by a lot of presidents in the past. The only saving grace regarding this project is that the method of capturing the signature is much different in the autopen compared to this project. The autopen has special cards that it can read off of that contain the location of the strokes and is specifically built for signatures [37]. Our machine's purpose can be extended from signatures to generic handwriting samples and this gives a small potential for being patentable even with the autopen patent.

Detailed Technical Description of Project

Three major components were essential to creating the automatic signature machine. First, we needed to design a secure point-to-point software system that can accept inputs from a user and subsequently convert them into logical motion commands to replicate the signature. Second, we needed to build a robust electrical system that can convert motion commands into physical electrical signals to drive the motors. Third, we needed to build the physical machine to execute the instruction exactly. In this section, we are going to present a detailed technical description of how we designed and delivered the project. Appendix A contains a block diagram that shows the entire system.

A. Secure point-to-point software system

The main responsibility of this system is to convert user inputs into motion commands. There are three requirements for this system: to be secure, reliable, and fast. To fulfill the requirements, we proposed the following human-machine interaction flow diagram:

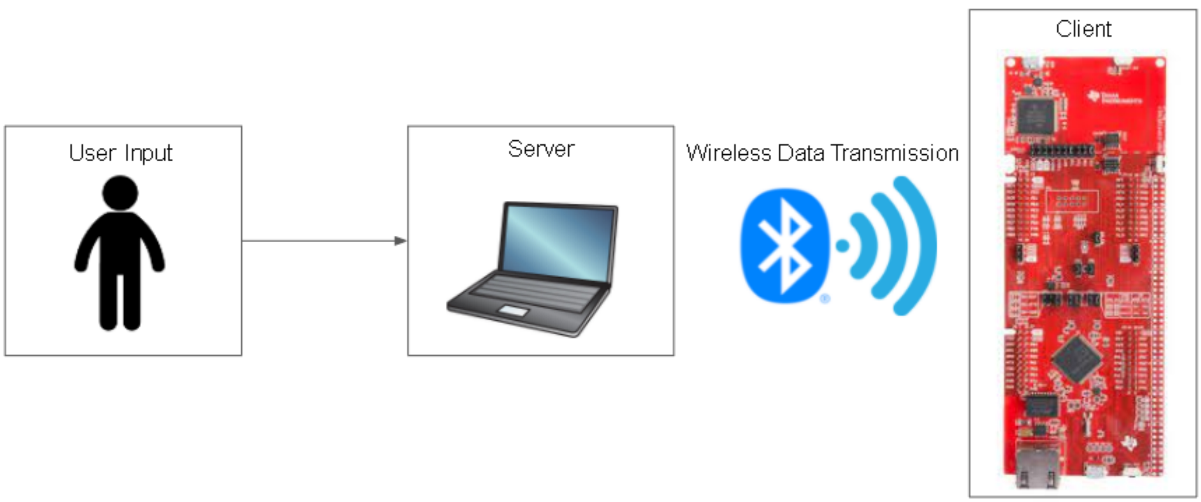


Figure 1: Human-machine Interaction Flow Diagram

As shown in Figure 1, there is a server that interfaces with the human and a client. The server can be an arbitrary entity, such as a remote cloud service, a desktop, or a laptop. This project will only consider the case for a laptop as the server. In the meanwhile, the client is a microcontroller that will establish a point-to-point connection with the server and continuously receive input streams. The microcontroller is responsible for parsing the input streams and converting them into logical motion commands. In this project, Bluetooth technology will be used as the point-to-point connection scheme because it is reliable and secure. Accordingly, the TI's MSP432E401Y is chosen as the microcontroller because it is Bluetooth compatible and powerful enough to compute the inputs into motion commands.

a. The Server

The server will accept two different forms of user inputs. The first is an image file that the user can place in the folder of the python script and run the program. This will cause the program to convert this image into coordinates which will be stored in a file and then another python script can be run to send these coordinates to the microcontroller via Bluetooth. The other valid input is through a python program application that has a blank canvas in which the user can draw anything on the canvas and then press the send out button to have the machine replicate the drawing. These two methods would allow the most versatility in the user wishing to use this machine.

To go into the image file replication process in more detail. This image processing algorithm takes an image file as input with the user placing this image file in the same folder as the python script. Image processing takes the input image of the signature and transforms it into coordinates/steps so that the device can replicate the signature on paper. The first step in this processing will be to use a thinning algorithm to thin the signature down to a one-pixel length version [38]. The thinning algorithm is shown in the figure below.

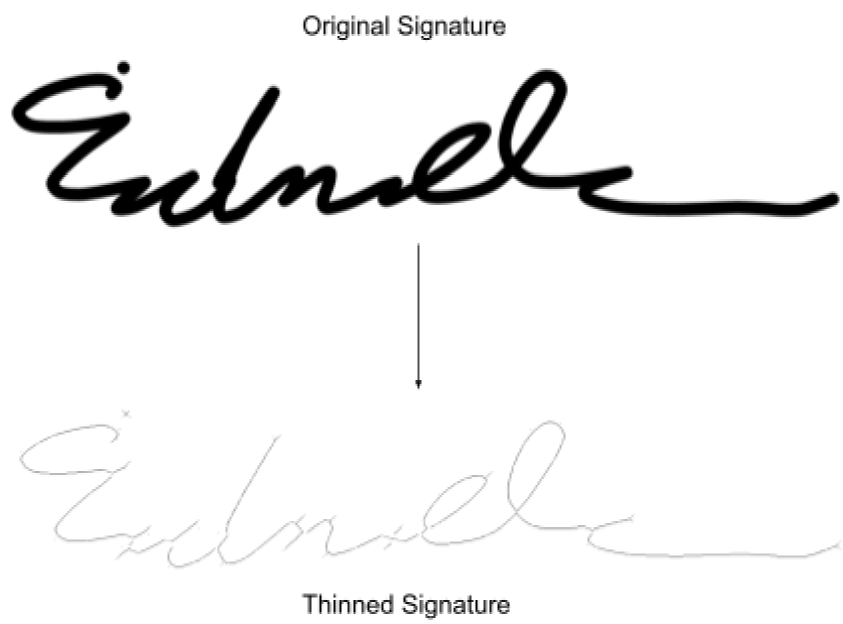


Figure 2: Image Thinning Algorithm

After the signature was thinned, there were a lot of stray pixels that were leftover by the image thinning that would cause major issues when replicating the signature, so a denoising filter was used [39]. The denoising filter that was used was the fastNlMeansDenoising that was provided in the python OpenCV module. This filter was able to successfully remove the stray pixels at the cost of a blurry picture. However, since the pixels of the picture were still able to be obtained, the blurriness of the picture is a nonfactor when deciding on what picture to choose.

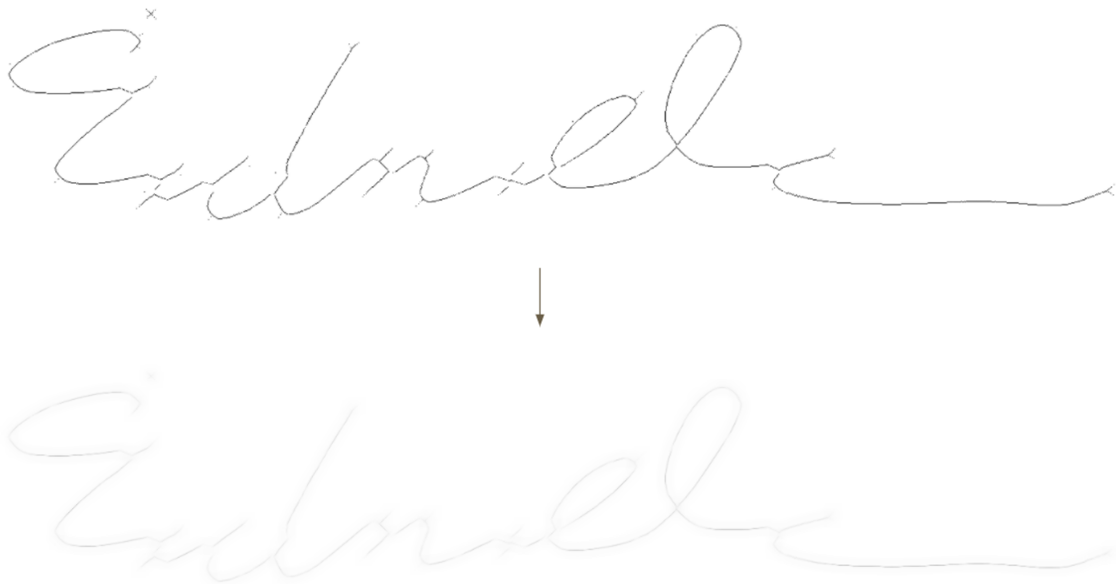


Figure 3: Image Denoising Algorithm

Figure 4: Image To Pixels

After these pixels were obtained from the image after denoising, then the strokes would have to be replicated and this was done by creating a minimum spanning tree using the pixels as vertices and the distance between each pixel as the edges. The minimum spanning tree would cause the shortest edges to be connected and essentially replicate the signature strokes for us.

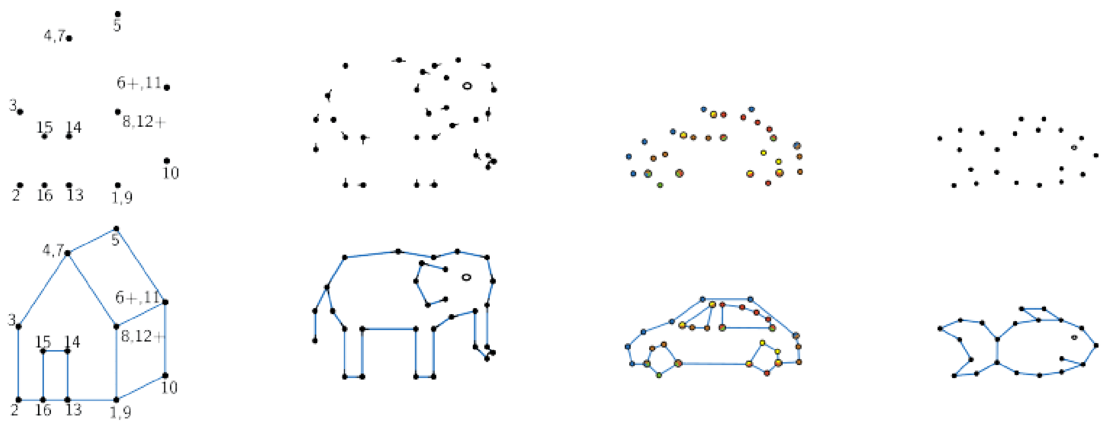


Figure 5: Connecting the Dots

Then afterward all that is needed is to determine the order of the coordinates to send to the microcontroller which can be done by just following the edges of the minimum spanning tree and sending the coordinates in that order after choosing an arbitrary starting point. These coordinates were then sent via Bluetooth to the microcontroller using the python Bleak module [34].

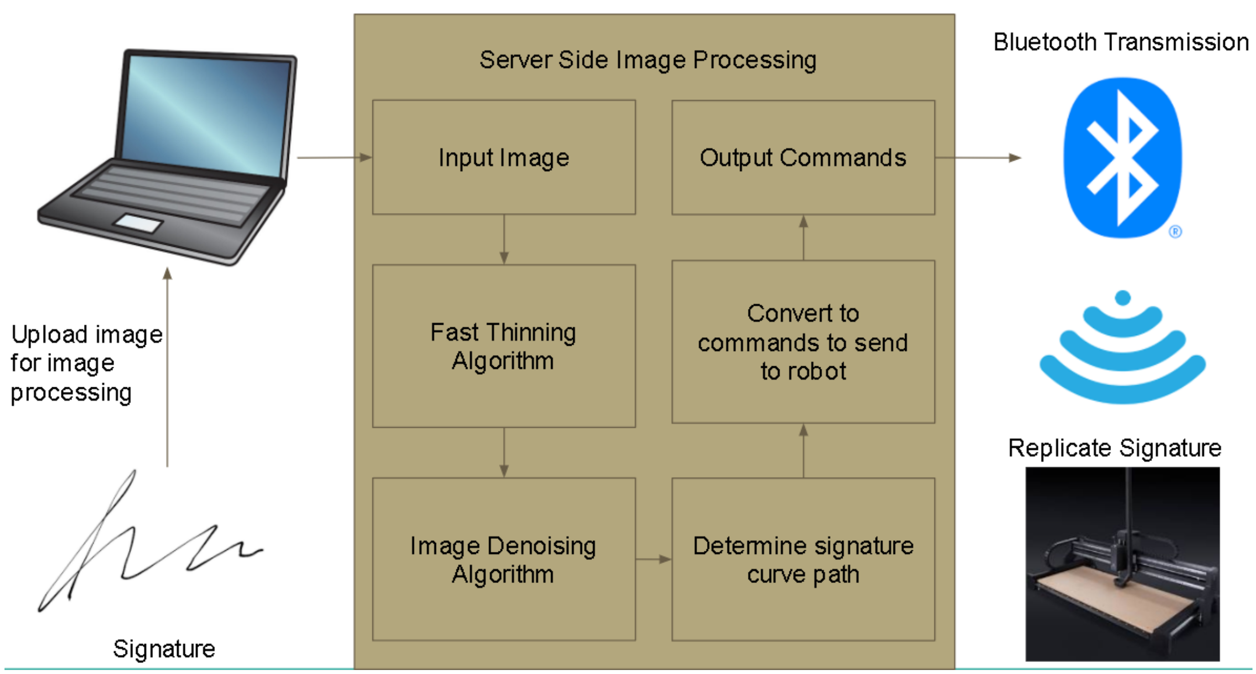


Figure 6: Image Processing Algorithm Diagram

The second method to accept the user input is to let the user draw on a digital canvas. We built a python app using a library called Kivy to create a user-friendly interface [40], as shown in Figure 7.

Figure 7: Kivy Application

This app is able to run on a laptop with touchscreen capability, such as Microsoft Surface Pro 4 [41]. When the user touches the screen of the canvas, the canvas is going to record the position of that interaction and draw the black line/dot accordingly on the canvas. The data will be stored in the coordinate format discussed later. There are three major motion capture functions: on_touch_down, on_touch_move, and on_touch_up. Together, this captures all the possible interactions that the user can have with the screen. Subsequently, the app provides the user with two options: Clear and Sendout. On the Sendout command, the recorded data will be packaged into ASCII bytes and sent to the machine via Bluetooth connection (please refer to the following section). On the Clear command, the existing data will be cleared, so the user will have a fresh canvas to start a new drawing.

b. Point-to-point Communication

To host a Bluetooth Low Energy (BLE) server on the microcontroller, a CCS2650 Module Booster pack [42] was used along with the sample code of project_zero given on the TI website [43]. This would allow the microcontroller to store and receive 40 bytes of data that could be accessed via memory by the microcontroller and sent by the python script using the Bleak module. This was the method of transferring coordinates between the server and the microcontroller.

Bluetooth was chosen as the method of wireless communication instead of other alternatives such as WiFi because Bluetooth Low Energy was easily supported using a booster pack and a sample code was already provided on the TI website. It was also chosen because of its simplicity and the project only required local wireless transmission which Bluetooth is perfect for. While there may be some transmission errors and packet loss during Bluetooth transmission, the Bleak module ensures reliable data transmission.

When the coordinate was sent to the microcontroller in the form of a command, it took the form of "(x,y)", where x is the x coordinate and y is the y coordinate in terms of pixels on the screen. In the callback function that was called whenever the microcontroller received a coordinate, the coordinate was parsed and the x and y coordinates were stripped from the coordinate sent from the server to the microcontroller. This format was chosen as it was simple to differentiate between multiple coordinates in the same message using the parenthesis to distinguish between multiple coordinates. It is also easy to parse the x and y coordinates by using the comma as a delimiter. The number of coordinates sent in a single Bluetooth transmission could be optimized by sending multiple coordinates at once, but through experimental verification, the signature replication time is bottlenecked by the actual motor movement. While this optimization could be future work, it would require optimization of motor movement speed before this optimization would be effective.

To have the ability to use multiple commands such as "Start", "Lift Pen", and "Finish", a command class was created which stored two 16 bit integers for x and y coordinates respectively and a command_t type to store the nature of the command whether it was a START, LIFT_PEN, COORDINATE, or STOP command. In this callback function, this command type was created and added to a queue that would store the coordinates for processing by the microcontroller.

c. The Microcontroller

When developing software for the microcontroller, the TI real-time operating system (RTOS) was used as multiple processes could be run, both the Bluetooth server hosting process and the coordinate process that will be gone more in-depth into later. Other benefits of using the TI RTOS are a large number of drivers that allow much easier programming of input and output ports along with timers and PWM pulses. GPIO pins could be mapped and configured as necessary using the GPIO.h file in the TI RTOS source files. Timers could be configured as necessary using the Timer.h file.

The software architecture of the microcontroller was to have one thread on the operating system handle the Bluetooth server hosting and the addition of these commands that were sent from the server to a local queue. Using the producer-consumer pattern [44], one thread was able to add coordinates to the queue and another thread was able to process these coordinates and dequeue them as they are added.

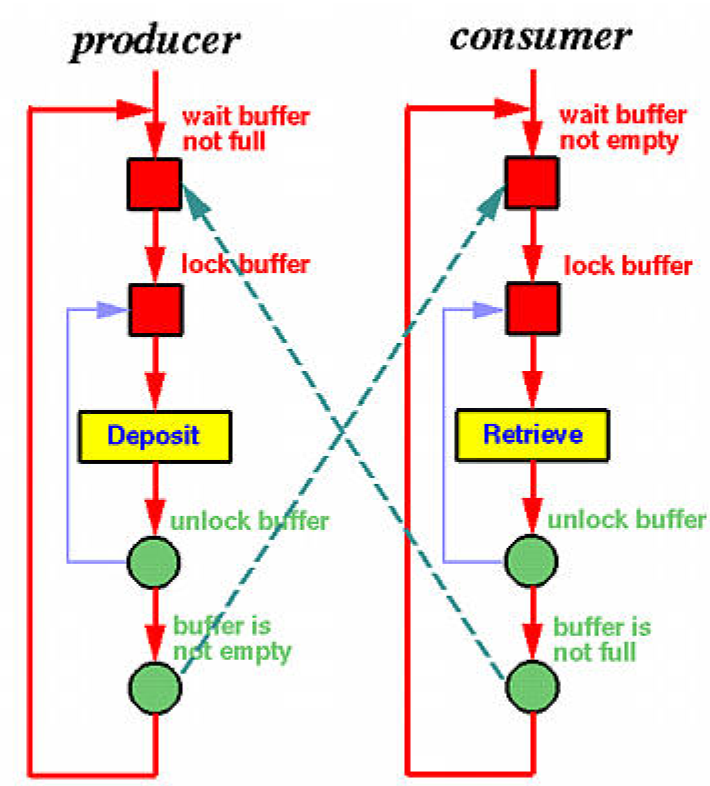


Figure 8: Producer-Consumer Pattern

A custom queue implementation was created to use for coordinate processing using the command struct. An array was used for implementation in the backend as a linked list implementation would come at the cost of memory usage. The array was used as a circular buffer which would store the coordinates until they were processed.

Making sure that the queue is not empty before it is dequeued, locks were used to stop the command processing thread until a command was ready to process. Similarly, if a command was currently being dequeued, the callback function was put on hold to make sure multiple threads do not access the same resource at the same time.

To process the coordinates, the idea behind the algorithm was that the microcontroller will output pulses to a motor driver that represents the movement of both the stepper motors controlling the x and y directions. Since each pulse will trigger a step increment on the stepper motor, the speed of the stepper motor can be controlled by changing the intervals between the pulses. To create the straight line between the two coordinates, the relative speed between the stepper motor in the y-direction and the stepper motor in the x-direction will be set equation to the slope. The pen will move from the first coordinate to the second coordinate while drawing a straight line. The base speed was experimentally found to be 500Hz as fast speeds resulted in lower quality drawings.

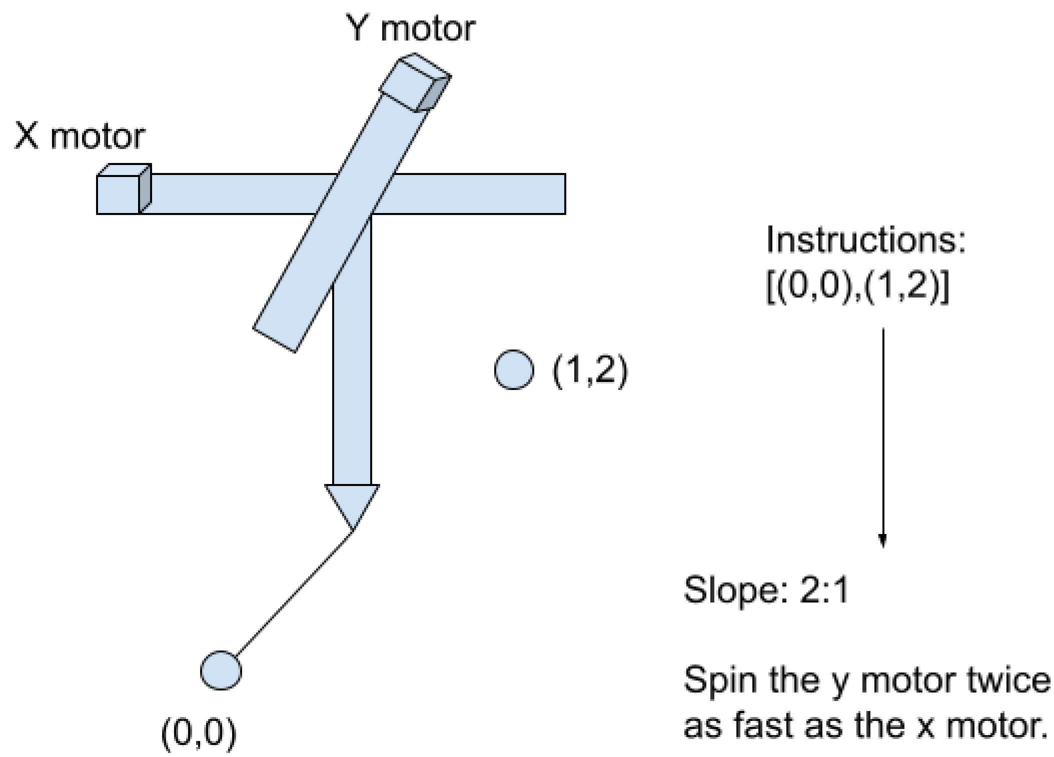


Figure 9: Coordinate Processing Algorithm

When actually implementing the algorithm, there were a couple of challenges. Since the stepper motor had to move accurately and precisely, one major issue was that the stepper motors could only move at certain speeds. After manual testing, it seemed that it was only really viable to step from 500Hz to 1kHz. Faster speeds ripped the paper that the signature was being written on. Since our stepper motor driver allows up to 1/16th micro-stepping, the largest magnitude of slope that could be used is 32 and the smallest was 1/32. The design decision made was that if the pen was lifted and not drawing, the command that would result in a slope that is out of bounds was split into two different commands that moved in the x and y direction independently. If the pen was down and drawing, the decision was that a straight line was drawn if the slope was too large and a horizontal line was drawn if the slope was too small. This introduces error into our algorithm and signature replication but this error is one that we were willing to accept as this was a constraint of the motor choice.

To send pulses or steps to the stepper motors, a timer interrupt was used that utilized the algorithm above to determine how fast the pulses should be. There were two timer interrupts that each sent pulses to each of the two motors. There was a separate timer interrupt that determined the length of time that the motors each ran their respective speeds. Through this method, the strokes of the drawing could be replicated successfully.

To handle the "Lift Pen" command, the solenoid was controlled by the microcontroller. Every time the "Lift Pen" command was processed, it would lift the pen by sending a logic signal to the bipolar junction transistor described in the PCB section. Whenever the next coordinate is processed, a logic high signal is sent to lower the pen and continue drawing.

As a safety mechanism, four switches in all the cardinal directions were added as an emergency failsafe. If there was something wrong with the motors or a bug was somehow introduced at runtime, these switches defined the outer bounds of the motor and when pressed, it would trigger an interrupt on the microcontroller that would stop whichever motors were running.

As mentioned earlier, there were multiple commands that were supported by the microcontroller. The first command is the START command in which the microcontroller sets an origin point for the start of the replication process. To do this, both the bottom left switches are hit and then the microcontroller is hardcoded to start at a certain portion of the paper on the base. This command is required to correctly set the origin point of the drawing and to ensure consistency in multiple replications. The COORDINATE command just signified a coordinate and that the coordinate processing algorithm listed above should be used. The LIFT_PEN command signifies that the pen should be lifted until after moving towards the next coordinate specified. The FINISH command specified that the drawing is now finished and then moves to the origin point that was initially set by the START command. If no START command was sent, then it moves to where it believes the origin is, unless stopped by the emergency failsafe switches.

The pins that were used for all logic inputs regarding the processing of coordinates and commands were determined by the PCB layout which will be described in much more detail in the next section.

B. The Robust Electrical System

It was crucial to have the appropriate motors for the machine to accurately replicate the signature. In this project, we choose two stepper motors to perform the x-axis and y-axis movements of a pen. Stepper motors were chosen for this project because they can move and stay at a known position [45]. They are known for their reliability, accuracy, precision, and consistency [46]. We also choose to use a solenoid to lift the pen upon receiving instruction because the solenoid is robust and simple to use. Four switches are chosen to be mounted at the end of the machine to detect any collision and prevent motors from moving out of bounds. Finally, we designed a power management system to distribute enough energy for the motors, and the PCB was designed to connect all the motors and switches with the microcontroller and power system. The following sections will discuss the detailed design choices made during the electrical design.

a. Stepper Motor

A stepper motor is a brushless DC motor that rotates in steps [47]. It is a permanent magnet surrounded by the windings of the stator. There are four wires that drive the stepper motor, and each wire will have the same pulse but at different phases.

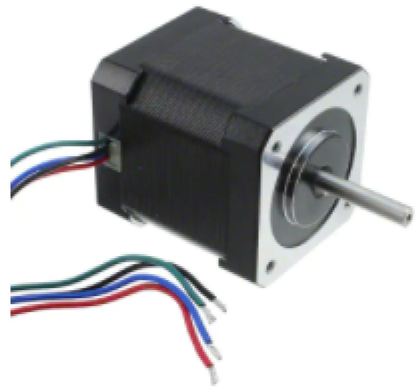
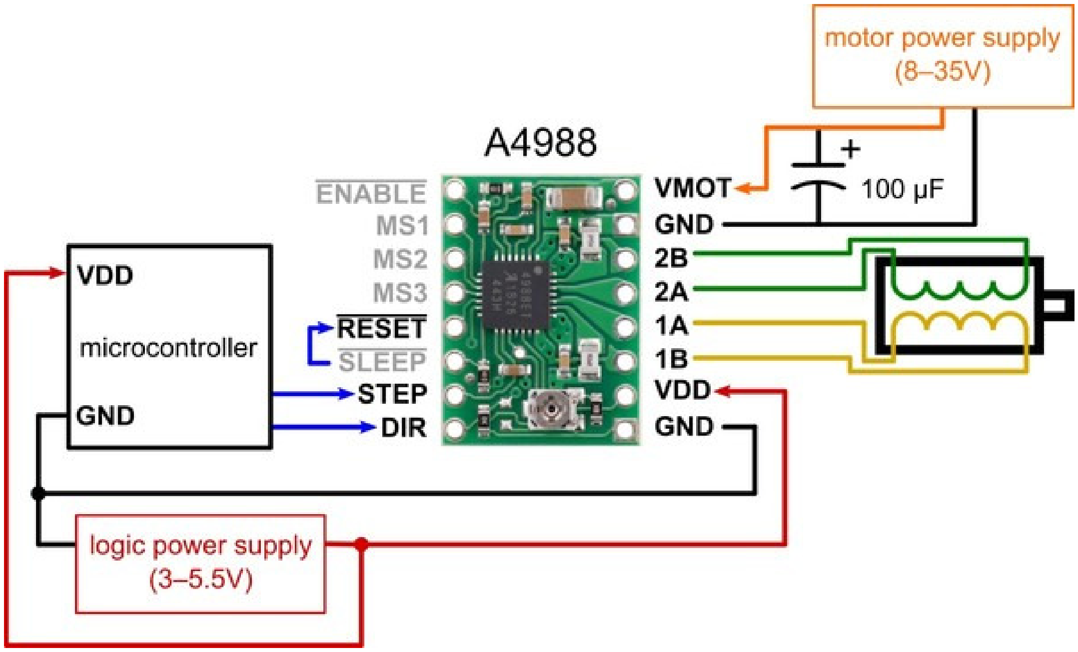


Figure 10: Stepper Motor

In this project, a NEMA 17 stepper motor, as shown in Figure 10, is selected to move the pen in the x-axis and y-axis [48]. The motor has a rated current of 1A and a coil resistance of 5 Ohms. It has a 490 mNm torque, so it has enough force to drive the weight of the load (please refer to the mechanical section for detailed calculation). The motor also has 200 steps per revolution, which is accurate enough to draw the signature (please refer to the mechanical section for detailed calculation).

In order to control the motor, we chose the A4988 driver chip [49] carrier as shown in Figure 11. The carrier [50] has a simple interface that exposes only the step and direction ports to control the motor. It can accept 8 to 35V of input voltage to power the motor and also works with the 3.3V logic control voltage from the microcontroller. In addition, it has 8 configurations for different micro-stepping resolutions. This is controlled by the MS1, MS2, MS3 ports as shown in Figure 12, 13. All of these make the carrier an ideal choice for the project.

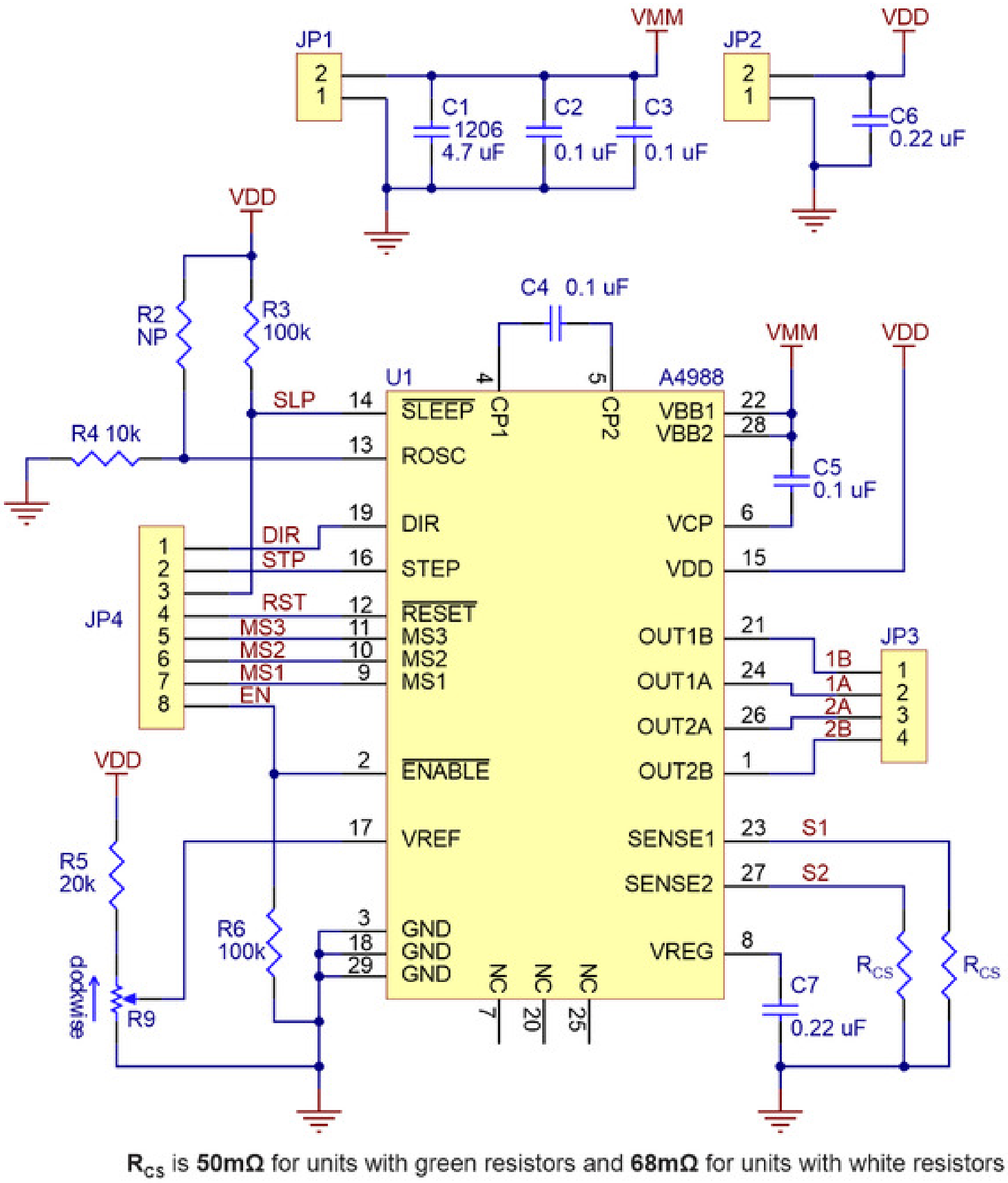


Minimal wiring diagram for connecting a microcontroller to an A4988 stepper motor driver carrier (full-step mode).

Figure 11: A4988 Stepper Motor Driver Diagram

MS1	MS2	MS3	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step
High	High	Low	Eighth step
High	High	High	Sixteenth step

Figure 12: A4988 Stepper Motor Driver Microstepping



Schematic diagram of the A4988 stepper motor driver carrier (both green and black editions).

Figure 13: A4988 Stepper Motor Schematic

We designed the schematic to interface with the driver carrier as shown in Figure 14. The input voltage is chosen to be 24V because the recommended voltage of the stepper motor is 24V and increasing the voltage generally allows for higher step rates and stepping torque since the current can change more quickly in the coils after each step [50]. There is a 100uF electrolytic capacitor [51] across the motor power and ground close to the board because the carrier board uses low-ESR ceramic capacitors, which makes it susceptible to destructive LC voltage spikes, especially when using power leads longer than a few inches [50]. As indicated on the datasheet, the RST pin is floating. Since we are not using the pin, we just connect it to the adjacent SLP pin on the PCB to bring it high and enable the board.

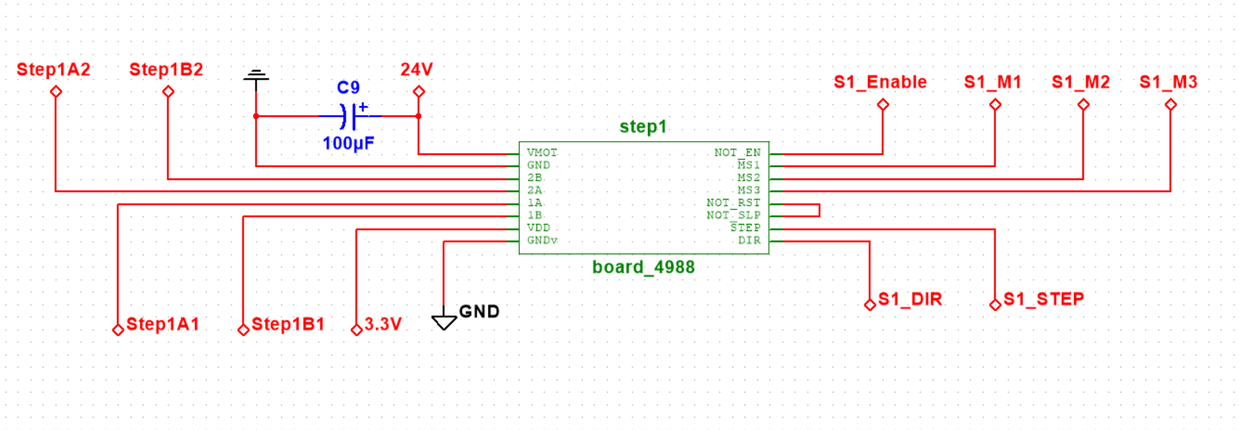


Figure 14: PCB Schematic of Stepper Motor

Lastly, the physical layout of the wiring is shown on the snapshot of the Ultiboard design as shown in Figure 15. Note that all the trace lengths are calculated to be 30 mils, which is sufficient to endure the 1A current flowing through the board [52].

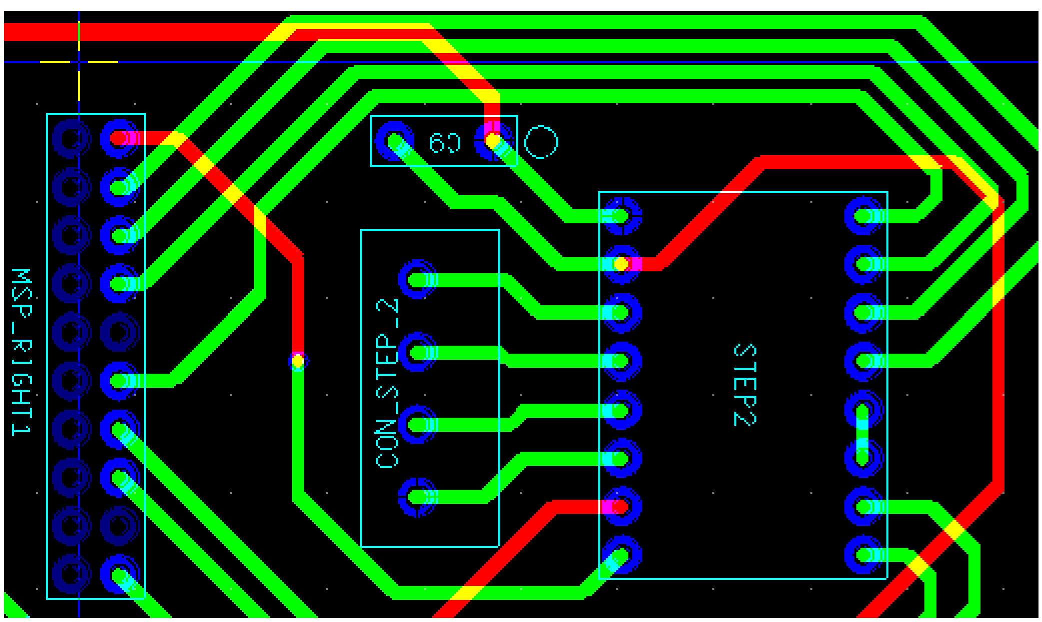
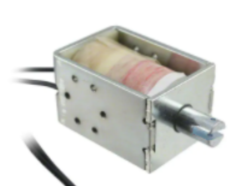


Figure 15: Ultiboard Traces of Stepper Motor

b. Solenoid Motor

Figure 16: Solenoid Motor



A solenoid motor is a device that converts electrical energy into mechanical energy by passing an electric current through a wire loop contained within a magnetic field [53]. It is robust and simple to use. In this project, a continuous pull solenoid is chosen to lift pens when necessary. Specifically, as shown in Figure 16, the solenoid was purchased at Digikey with product number F0461A from Pontiac Coil Inc. This solenoid has a stroke length of 1.25" and can provide a 5 oz force, which is capable of lifting a pen from the writing paper. This solenoid is rated for 13W with a 12V rated DC and 11 Ohm DC resistance [54].

Because this solenoid has only the ON and OFF states, we made a design decision to use a BJT as a switch to control the solenoid. It is beneficial to use a BJT as a switch in this project because BJT is simple to use and easy to prototype. In contrast to a solenoid driver chip, it drastically reduced the designing overhead, because it had a much simpler datasheet to examine and a less intricate wiring structure that allowed us to create prototypes. As a result, an NPN through-hole BJT with the Manufacturer Product Number as LM395T/NOPB from Texas Instruments was selected [55]. As shown in Figure 17 this BJT is rated for 36V and 2.2A, which is capable of handling the power going through the solenoid. More importantly, this BJT has an internal protection circuitry, as shown in Figure 18, which can safeguard the circuit in case of a shortage from the collector to emitter.

Figure 17: BJT

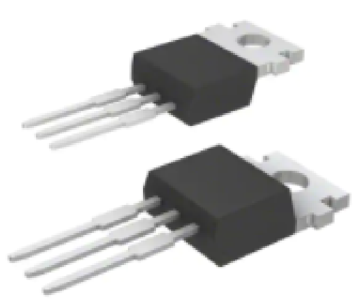


Figure 18: BJT Schematic

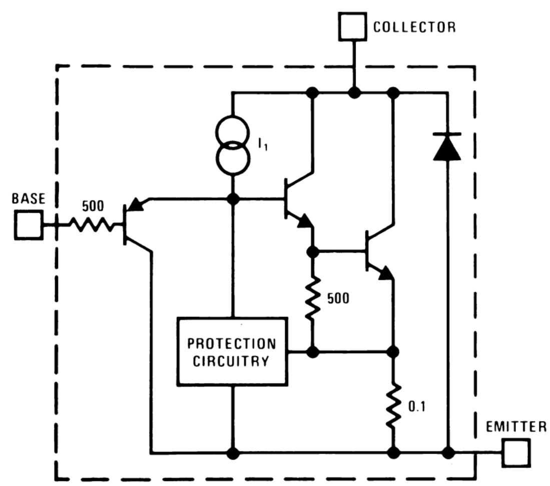
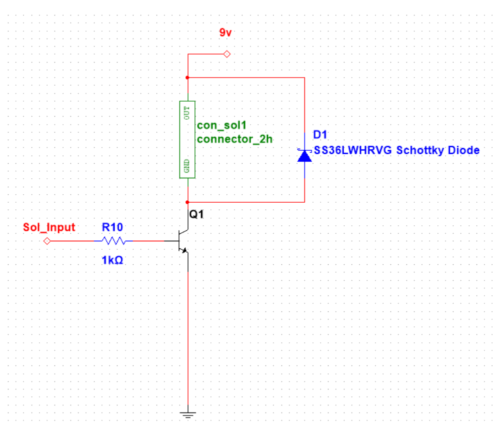


Figure 19: Solenoid Circuit Schematic



Subsequently, we designed the schematic as shown in Figure 19. Since the purpose of the BJT is to serve as a switch, it should not matter whether the load is placed on the emitter side of the collector side. However, it is only possible in this project to have the solenoid on the collector side. Otherwise, as shown in the equation below, the base is connected to the logical input which can maximally supply 3.3V and the solenoid has a resistance of 11 Ohm, which, as a result, cannot supply enough current for the solenoid to work properly [56].

$$IC \approx \frac{VBias - 0.7}{RE}$$

In the meanwhile, a 1k Ohm resistor was selected to interface the logical input to the base of the BJT to serve as a protection to limit the flowing current. Lastly, to prevent the reverse polarity voltage pulse on switch turnoff, a Schottky diode as a flyback diode is connected in parallel across the solenoid. The function of the flyback diode is to allow an electric current to pass in one direction only while blocking it in the opposite direction [57].

The Ultiboard layout design is shown in Figure 20. We used a two-position vertical terminal block from Phoenix Contact on the PCB board design to interface with the solenoid wires for testing purposes [58].

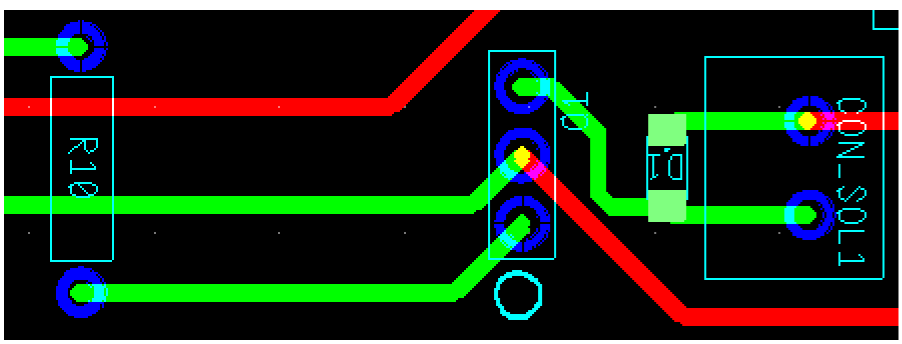


Figure 20: Ultiboard Schematic of Solenoid Circuit

c. Switches

Figure 21: Switch

A snap action switch by Omron Electronics Inc-EMC Div is selected in this project as shown in Figure 21. It can tolerate a rated 10A and 250V which is sufficient for the project.

Since the purpose of the switch is to detect any collision and prevent motors from moving out of bounds, the switch needs to connect to the microcontroller pins to trigger the GPIO interrupts. However, it is not safe to directly connect the switch to the microcontroller because the length of the wires connecting between a switch and the microcontroller are long and there could exist unwanted electrical surges and interferences that could damage the microcontroller. Therefore, we added two layers of protection to the schematic design. As shown in Figure 22, we first included an RC circuit to serve as a low pass filter to filter out unwanted noises. Second, we added a TVS diode [59] by Semtech Corporation that has a 3.3V reverse standoff voltage to avoid potential electrical surges.

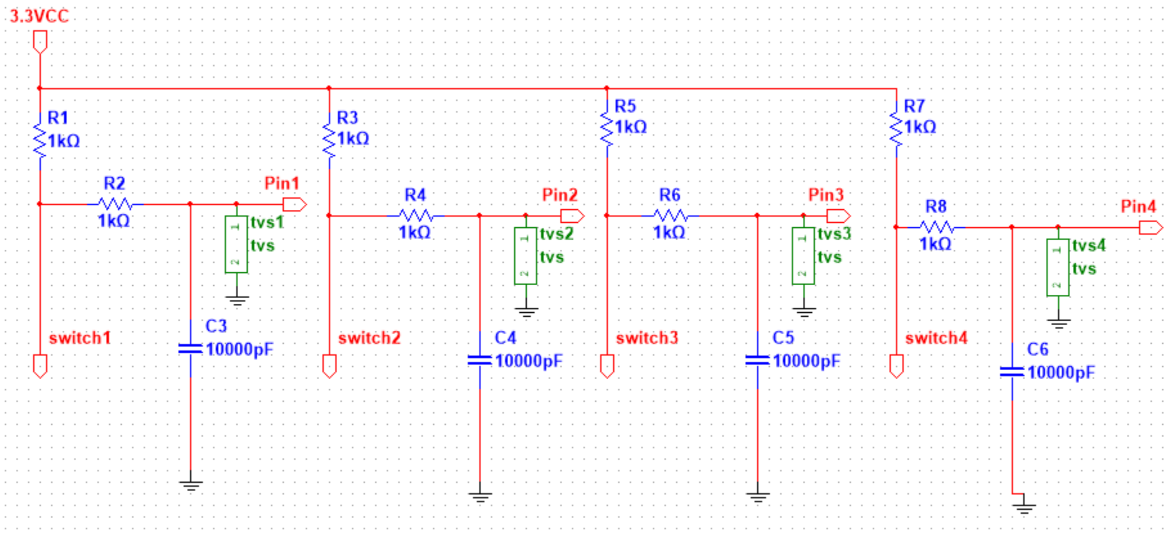


Figure 22: Schematic of Switch Circuit

Finally, we designed the Ultiboard layout as shown in Figure 23. We used a two-position vertical terminal block from Phoenix Contact on the PCB board design again to interface with the switch wires for testing purposes.

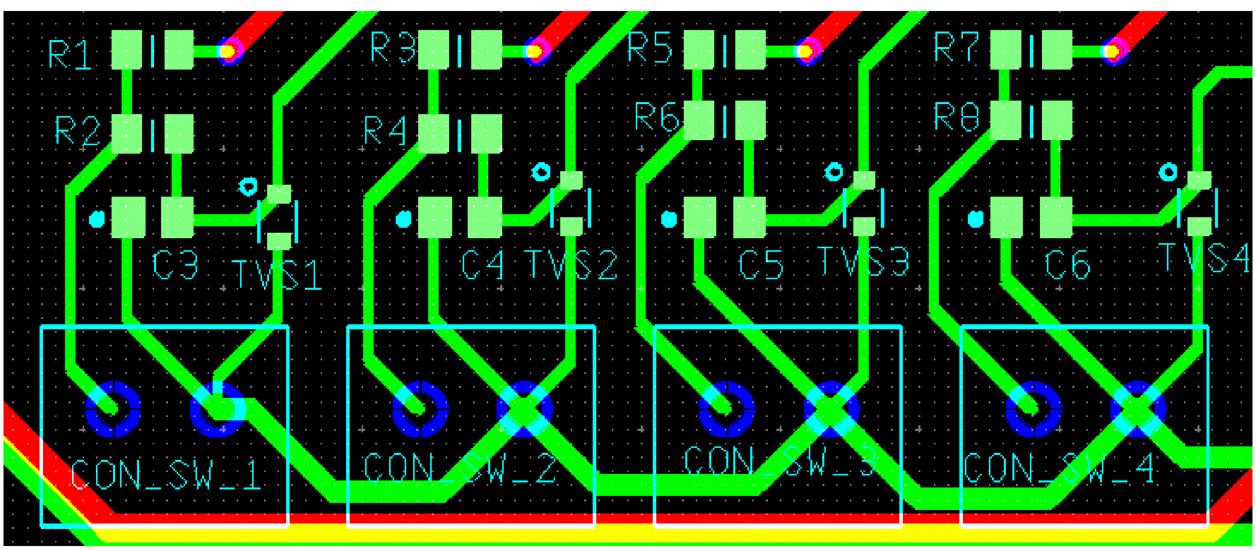


Figure 23: Ultiboard Trace of Switch Circuit

d. Power Systems

Figure 24: Power Supply

To supply the power to the microcontroller and the motors, we decided to use an AC-DC wall adapter to convert the standard 120V, 60Hz AC electricity to the desired DC output. Specifically, as shown in Figure 24, we chose the 24V, 160W AC/DC adapter by GlobTek, Inc. We chose 24V as the DC output because this is the standard DC voltage for the stepper motor as indicated on the datasheet [60]. Moreover, a higher DC voltage means a higher torque at higher velocity since the current can change more quickly in the coils after each step [50]. By calculation, each stepper motor and the solenoid will consume approximately 1A of current, and the microcontroller and switches together will consume no more than 0.5A of current. Therefore, the total current consumption will not exceed 3.5A. Then this 160W adapter suffices to provide all the necessary power to the system.

The power system can be divided into three parts because it needs to drive the stepper motors, the microcontroller, and the solenoid. As shown in Figure 25, there is a 24V output, a 3.3V output, and a 9V output. 24V output drives the stepper motor. 3.3V output drives the microcontroller and the switches. 9V output drives the solenoid. It is worth noting that instead of using 12V to drive the solenoid, we chose 9V because the experimental measurements showed us that this would work well for our system. Also, since the equivalent DC voltage for the solenoid was 11 Ohms and 12V would result in a larger than 1A current which is approaching the upper rating of the solenoid. The 9V and 3.3V outputs are generated by using a DC-DC converter by Recom Power [61] [62]. For extra protection of the circuit, a Schottky diode is added near each DC-DC converter. The diode in this project is chosen to be MBR360RLG by Onsemi. In addition, we added a few connector header pins for testing purposes.

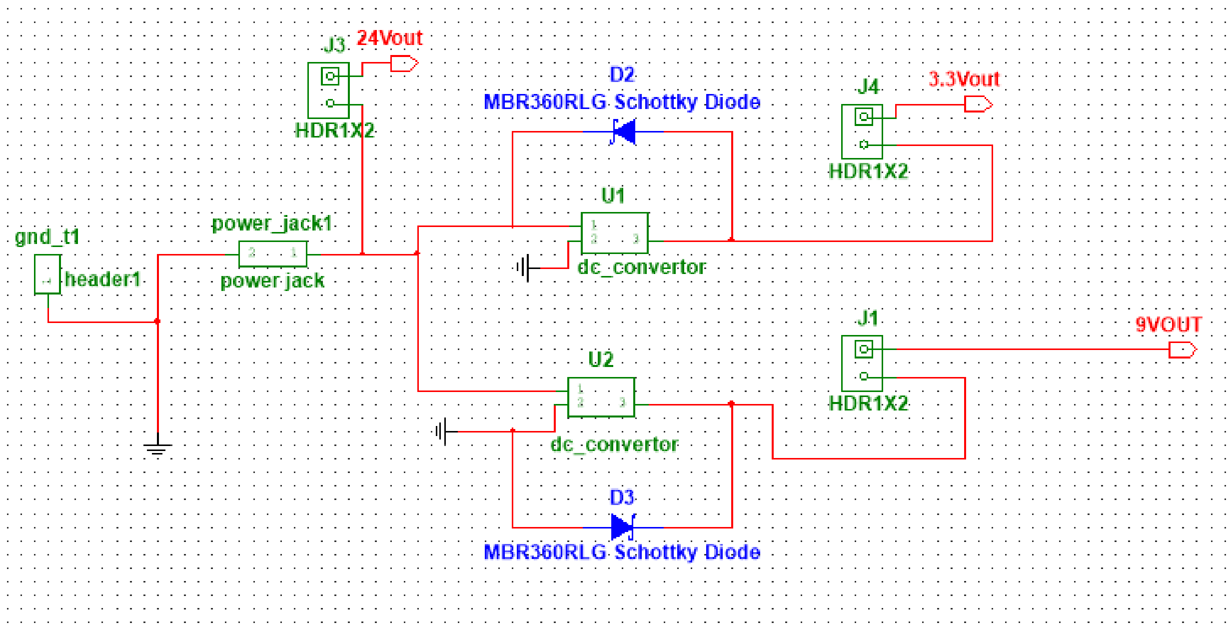


Figure 25: Schematic of Power Supply System

Finally, we designed the PCB layout on the Ultiboard as shown in Figure 26. We used several header pins, J1, J3, J4, to create an open circuit from the rest of the board layout. The purpose of this was to create a simple way to test our power supply using the test plan that is discussed later. Using these jumpers, the power supply and any additional tests can be done effortlessly and efficiently.

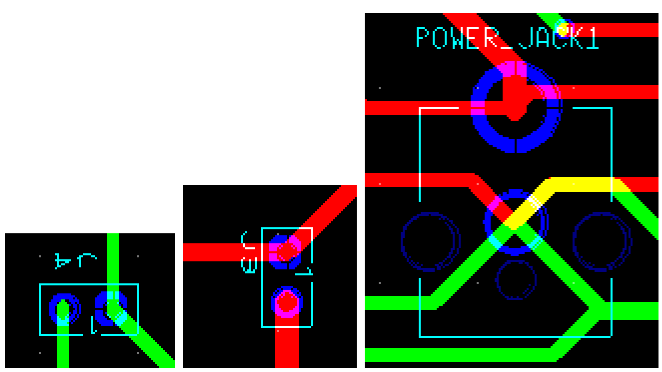


Figure 26: Ultiboard Power Supply Traces

e. Problems and Design Modifications

Initially, we were planning to use the DRV8436RGER stepper motor driver [63] and the DRV103U solenoid driver [64] by Texas Instruments to drive the stepper motor and the solenoid motor. We designed the schematic as shown in Figures 27 and 28.

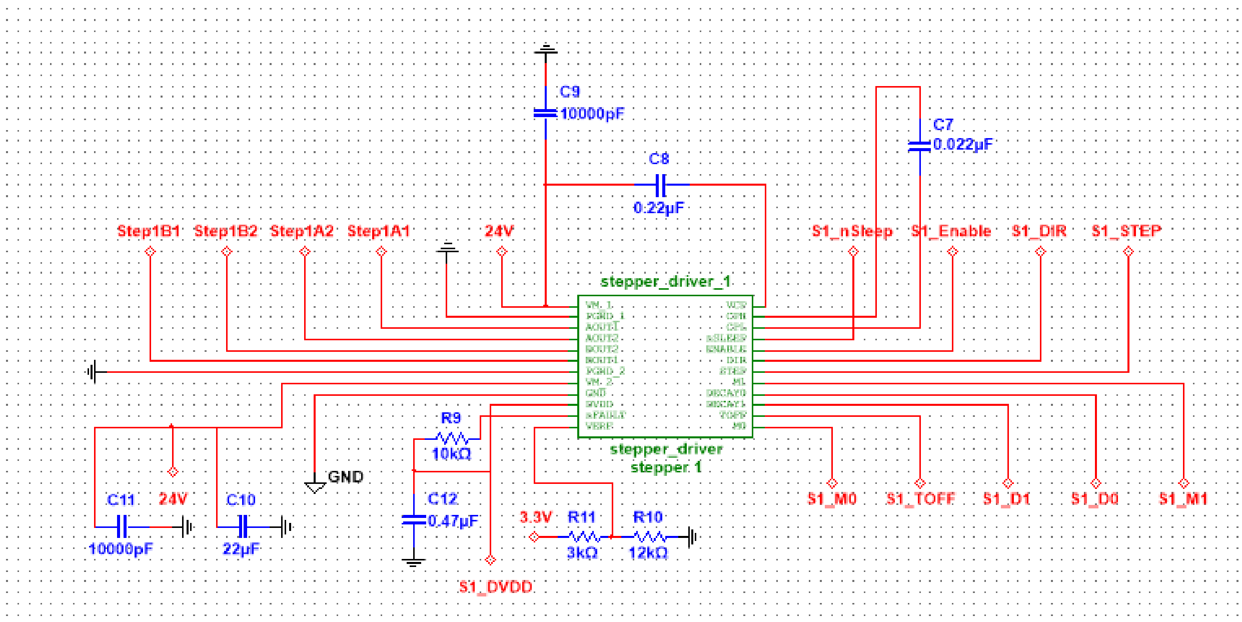


Figure 27: Stepper Motor Driver Circuit Schematic

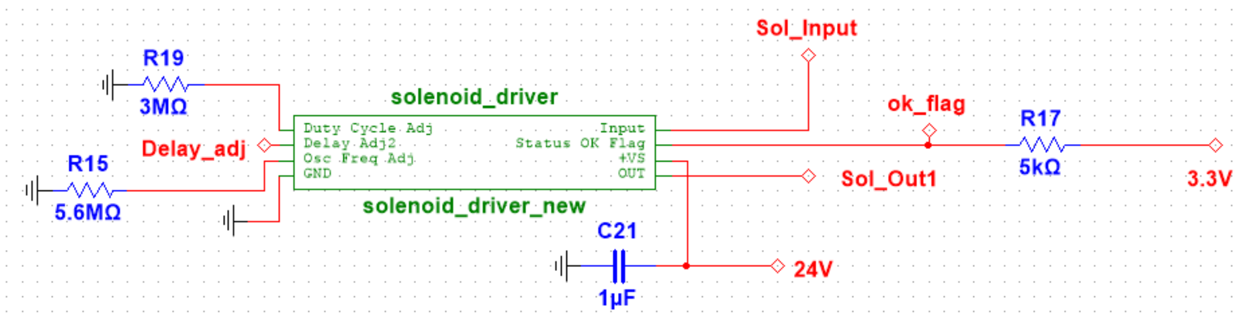


Figure 28: Stepper Motor Driver Solenoid Schematic

Unfortunately, during one of our test runs on the PCB, we accidentally shorted the board that caused the board to be ruined. However, because of the supply chain shortage issue during the pandemic, the DRV8436RGER stepper motor driver was now out of stock. It was impossible for us to create a new board without the parts and as a result, we were forced to relinquish our design and opted for plan B, in which we used the A4988 driver carrier to drive the stepper motor. In the meanwhile, we realized that the DRV103U solenoid driver was far more powerful than we needed, which significantly increased the complexity to manage the driver. As a result, we redesigned the solenoid controller and decided to use a BJT to act as a switch to control the solenoid. This significantly decreased the difficulty of managing the board. More importantly, it was clear to see that the complexity of the new design was much less than the old design, as shown in Figures 29 and 30.

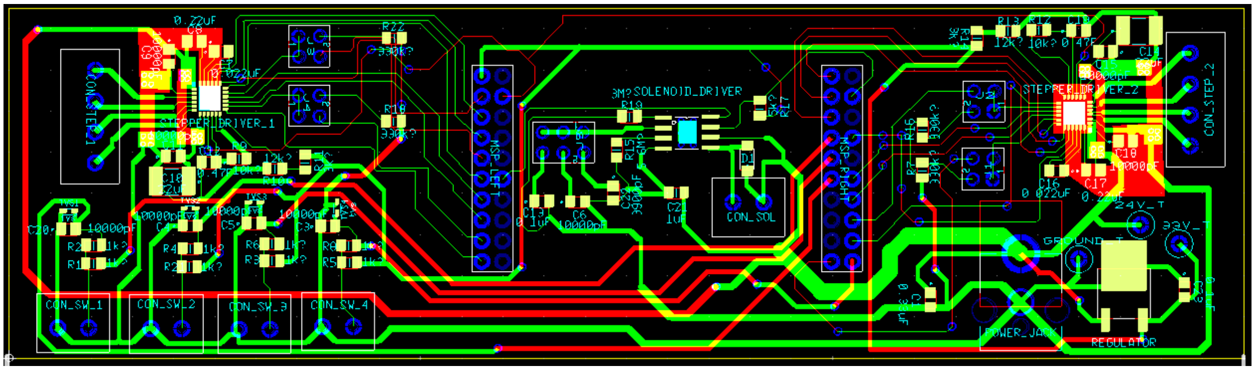


Figure 29: Old Ultiboard Printed Circuit Board

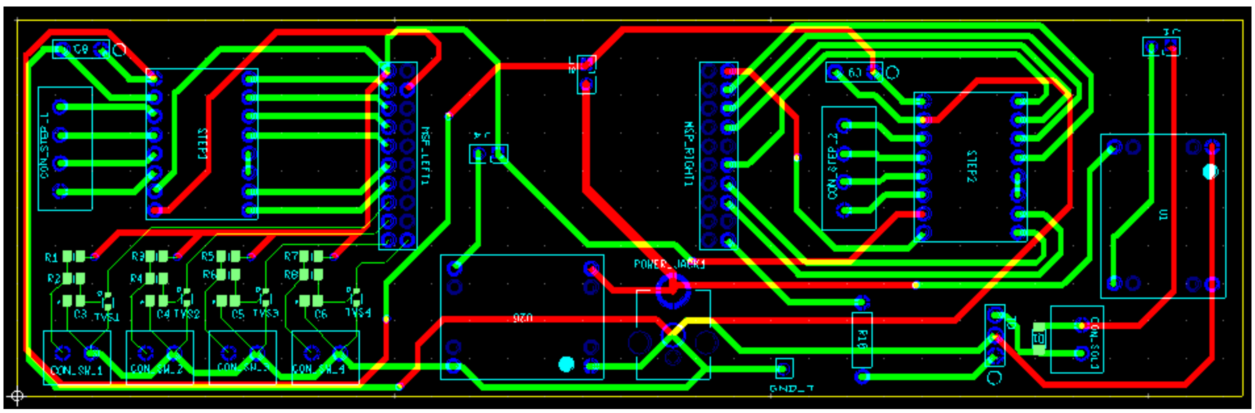


Figure 30: New Ultiboard Printed Circuit Board

C. Mechanical.

a. Source/References

The mechanical aspect of this project is divided into three major parts: movement of the pen, activation of the pen, and boundary detection. In order to install these mechanisms, a system framework must be developed. With the criteria of efficiency, resources, and pricing, OpenBuild was decided as the main source of materials for this design. Not only does OpenBuild specialize in 2/3 dimensional movements designs, its abundance of references, available Cad files, and overwhelmingly positive feedback from its community further establishes itself to the ideal market of choice.

b. Base Framework

For the system framework, the design is a combination of T-slots and V-slots aluminum extrusion. T-slots are used for framing and assembly purposes while V-slots are used for linear railing installation. As seen in Figure X, two V-slots are mounted parallel to each other. A gantry is installed in each of the extrusions, connected by another V-slot extrusion which has an additional gantry installed on top. Each gantry has 4 rubber bearings, creating 2 linear rail systems. These aluminum extrusions will be connected via cast brackets and L brackets. Based on the height of the V-slot extrusions and the height of the bracket, 10mm 32 thread screws and T-nuts are utilized to ensure proper connections. For a more visual explanation/labeling, see Figure 31.

Since the application of the machine is mostly on paper and similar writable material, to include flexibility for paper sizes, the design is determined to have an overall dimension of 25.875X17.438X10.625 inches in XYZ order.

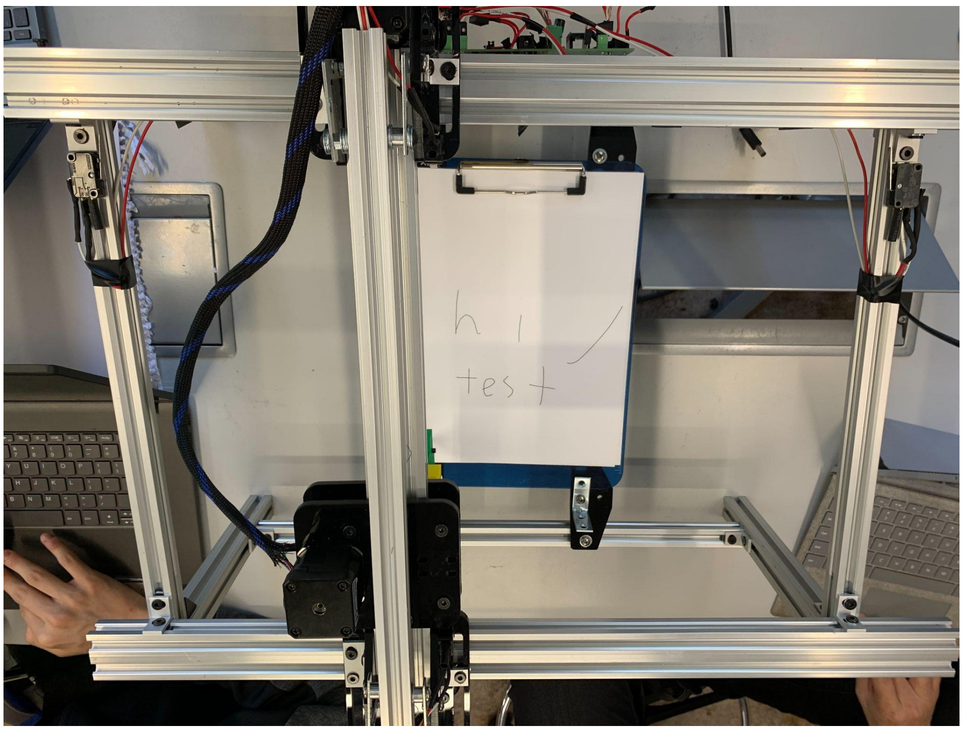


Figure 31: Design Framework

c. Movement Mechanism

To be able to have sufficient space for the motor installation without compromising the stability of the framework, 20X40mm sized V-slot extrusions were selected. To motorize the linear rails, a pinion belt system is used. On each side of the aluminum extrusion where the motors are parallel to, a 2mm pitch wide belt is attached while a pinion is screwed tight on the motor rod. The belt's teeth, facing towards the aluminum, go through the rubber bearings and are wrapped around the pinion teeth (Figure X). The ends of each belt are then secured with T-nuts and 32 thread screws. With this setup, as the stepper motor turns, the attached pinion is forced to rotate through the immovable belt, allowing the motor to turn teeth by teeth.

There were three constraints for the motor selection process: size, price, and torque. Since the Acro plate utilizes a NEMA 17 motor and its budget was given a maximum of $50 after purchasing most of the components, the only concern lies in the torque requirement. The main load of the design has a mass of around 1.216 kg. Based on the wire/conveyor belt system, the calculations for minimum torque are as follows [65]:

$$F = Force\ (N), \eta = Efficiency. i = Gear\ ratio, D = Pulley\ Diameter\ (m), F_E = External\ force, m = Mass, g = Gravity, \mu = Coefficient\ of\ friction\ (rubber)$$

$$F = F_E + mg(sin\theta + \mu \times cos\theta)$$

$$F = 0 + 1.216 \times 9.81\ (\sin(0) + 1.58(max) \times \cos(0)) = 18.85\ (\mathrm{max})$$

$$T_L = \frac{F}{2\pi \times \eta} \times \frac{\pi \times D}{i} = \frac{F \times D}{2 \times \eta \times i}\ (Nm)$$

$$T_L = \frac{18.85 \times 0.012}{2 \times 0.85 \times 1} = 0.133Nm\ (max)$$

Based on the calculations made, with a maximum coefficient of 1.58 from rubber to aluminum contact, the total torque comes to 0.133. Stepper motor model QSH4218 was selected, producing a maximum of .49 Nm, more than three times the required torque. This will ensure absolute consistent control of the motor while minimizing any possibility of gear slip.

d. Load design

The main load is considered the most constrained aspect of the mechanical design. This is where a pen is installed on the bottom and contains the y-direction motor on top. To fulfill the capstone's objective, the load must be able to control the activation of the pen and apply enough force for the pen to write on the paper. In addition, the load must account for potential changing heights as it moves along the rails.

Based on the mentioned criteria, a solenoid was chosen as the actuator for the pen. In particular, a pull solenoid was chosen over a push since directly pushing the pen down creates too much pressure which may damage the paper. Thus, a pulley system (Figure x) is incorporated in which forces the pen to push against its spring when the pull solenoid retracts. The pulley system for the load can be seen in Figure 32.

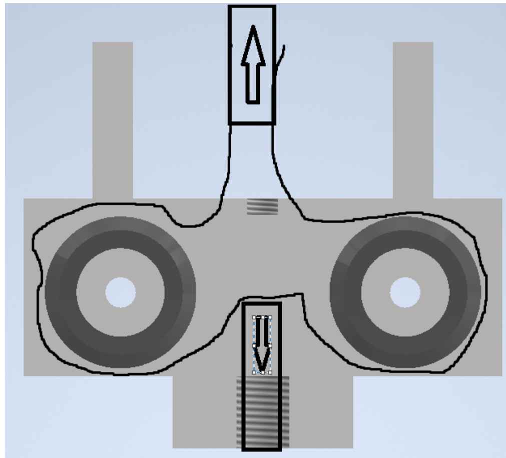


Figure 32: Pulley System

Since the shape of the solenoid was unique, 3D printing was required to customize the container onto the Acro plate. There are a total of two printed parts: the first to secure the solenoid below the plate and the second to hold the pulley system. The two parts are combined to become Figure 33.

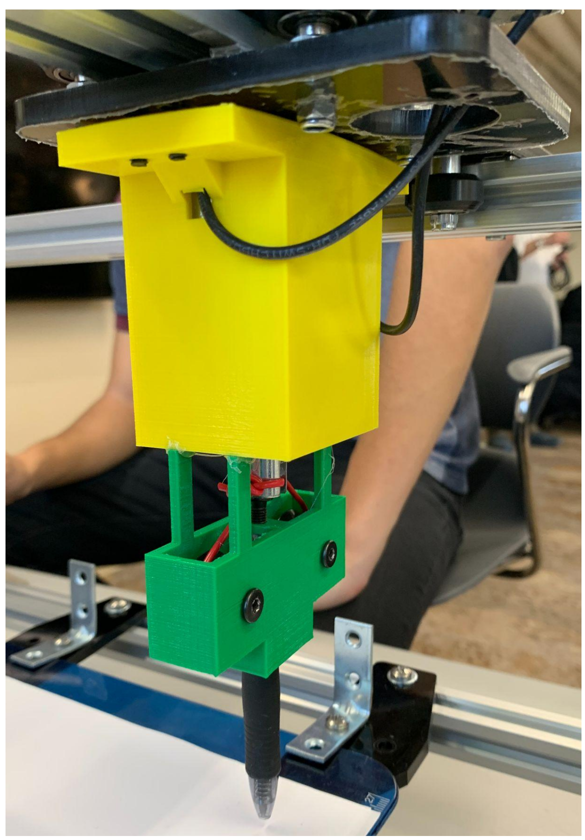


Figure 33: 3D Printed Parts

e. Wiring

A wiring of 20 AWG was used for all wire extension and mechanical component connections. To secure the wires and prolong their longevity from moving constantly along the linear rails, a sleeve was purchased to cover up the exposed wires.

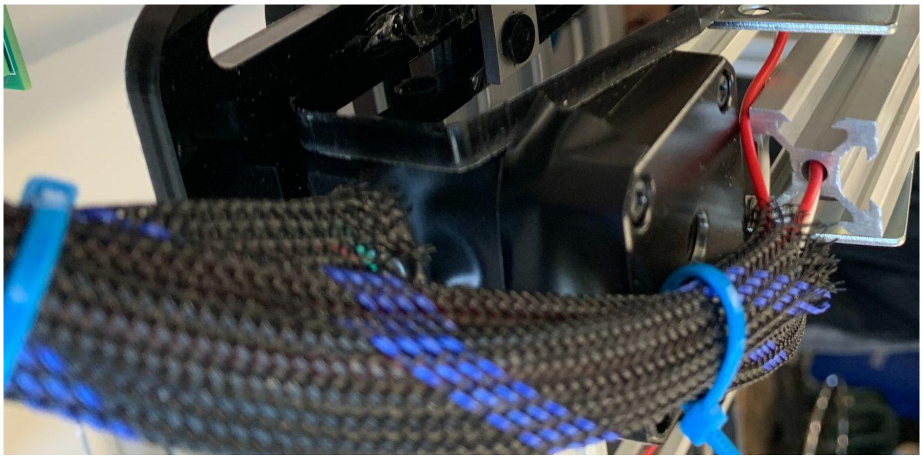


Figure 34: Wire Sleeves

Project Timeline

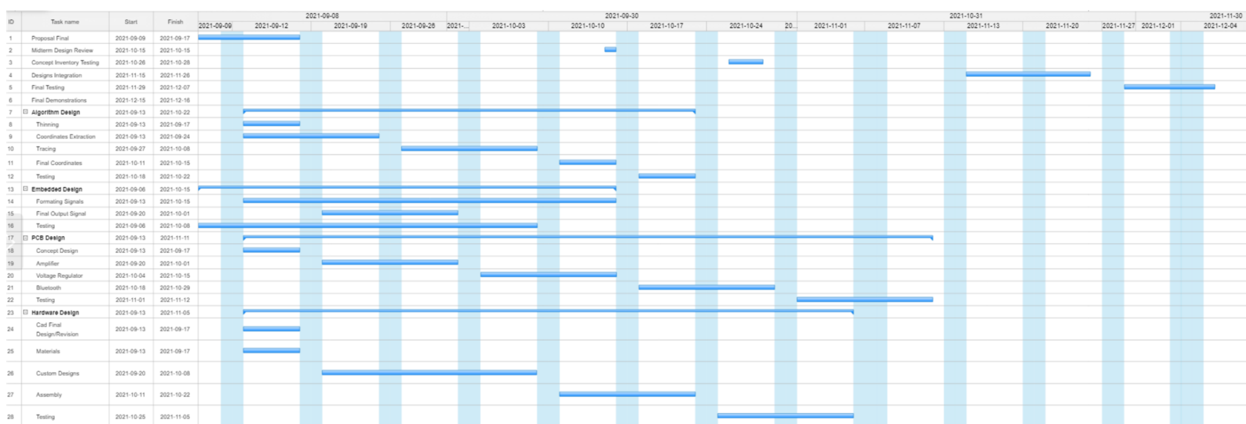


Figure 35: Gantt Chart Submitted in the Proposal

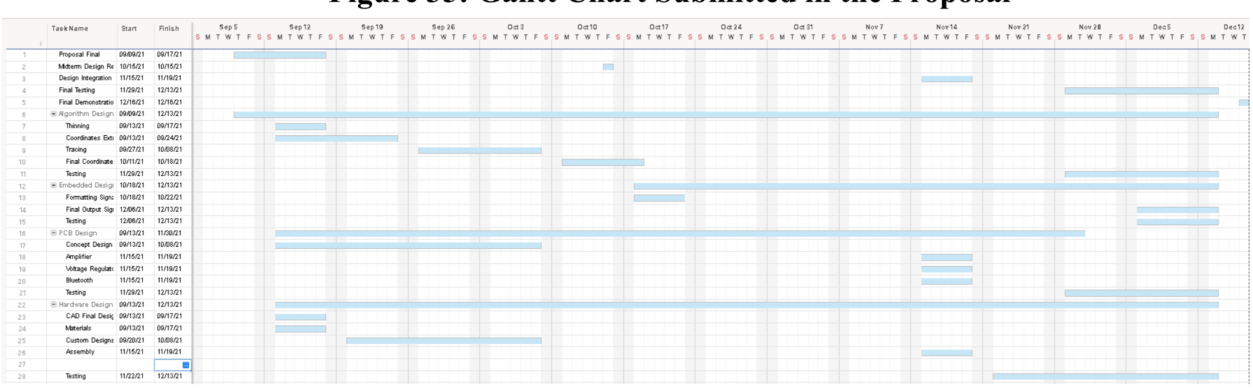


Figure 36: Gantt Chart Submitted in the Proposal

For this timeline, the tasks are mainly divided into four major sections (algorithm, embedded, PCB, and hardware), running in parallel with each other while continuing in series individually.

Initially, the tasks could be split up in parallel by working on the algorithm along with the schematic and the base. However, after all of these subsystems could be completed on their own in parallel, the integration was a serial task that relied on the completion of all of the other subsystems.

The work was divided into three parts in such a way where each person will have a primary role and a secondary role to back up another person's primary role. Yu-Jiyun's primary role is mechanical design, with a secondary role of algorithmic design. Zichao's primary role is embedded system design with a secondary role of mechanical design. Edward's primary role is the algorithm design, with a secondary role of embedded design.

We made the mistake of not frontloading the schematic and ended up near Thanksgiving break with not much time left and an incomplete project. Due to these long turnaround times of the parts, this drastically reduced the number of chances we had on getting the correct schematic and the team had to put a lot more effort during and after Thanksgiving break to get the project working successfully. Each of the subsystems was finished during Thanksgiving break and the integration and testing happened the next two weeks after.

The dates that were important that were discussed during the first day of class were the midterm design review and demo day. These deadlines were the most important to meet and we had to make sure we had the deliverables ready by those deadlines. The team's priorities changed near these dates to successfully meet the deadlines.

Test Plan

Our original test plan was to test both the hardware and software separately and then integrate them together afterward. On the software aspect, multiple different signatures were tested with different edge cases and the list of coordinates was printed out and stored into a text file. Now since these coordinates are enough to verify that the program works, so a separate testing program was made to simulate what the microcontroller would do. This testing program essentially took coordinates and drew straight lines between the coordinates with the exception of the "Lift Pen" command in which it did not connect the two coordinates separated by the "Lift Pen" command. This method of testing was used as this was exactly how the microcontroller should theoretically process these commands so it was essentially a simulation of the actual signature replication process.

To test the Bluetooth transmission and the Bluetooth server on the microcontroller, test programs were made to both write and read from a certain BLE GATT attribute. This involved writing a test string to the 40-byte memory location on the microcontroller and reading from that same attribute to see if the writing and reading worked properly. There were also breakpoints set in the callback function of receiving data from Bluetooth to also make sure the microcontroller was able to successfully receive the coordinate and call the callback function.

Once the PCB arrived and was populated, the first thing to test was the power supply system. There were test points on the board for three different power supply rails: 24V input power, 12V solenoid power, and 3.3V logic power. The input power was tested and successfully returned 24V, meaning that the wall adapter worked successfully. Both the 12V and 3.3V power were tested which meant that the two voltage regulators that were chosen also worked perfectly as well. Separately from testing power, it took some time to test how to supply external power to the microcontroller. Once this was figured out, there was integration with the power supply from the PCB and the microcontroller to power the microcontroller from the PCB from the wall adapter.

Separately from all of this, the motors were tested when the motor drivers came in. Using the NI VirtualBench to power both the motors and the logic power supply of the motor driver, function generators were used to simulate the stepping logic of the microcontroller and to test the functionality of the motors. Once this was successful, a similar process was done for the solenoid in which we tested input logic into a bipolar-junction transistor to test whether the solenoid could be turned on or off at will of the logic input. Once both of these subsystems were individually tested, they were individually added onto the PCB and tested as well. For example, after testing the motors individually, then the motors were connected to the PCB and microcontroller and test code was run to see if the motors would run off of the logic of the microcontroller. A similar process was done for the solenoids to successfully incorporate both subsystems into the project.

The last aspect of our project was the emergency failsafe switch mechanisms. The switches were individually tested to make sure the circuit we designed for them worked properly and once manual testing with the virtual bench was successful: logic high when not pressed and logic low when pressed; they were successfully integrated into the entire project as well.

The original test plan was to make sure each individual subsystem worked as intended and then integrate each small subsystem into the whole project one by one making sure that the integration did not cause any problems with the entire system. There were a couple of problems we did run into when testing our systems one of which was that the solenoid driver that we initially bought and designed our PCB for did not work at all. This resulted in building a new circuit that involved a simple bipolar junction transistor instead of the solenoid driver which was the one that was used in the final product. Another major setback was that the PCB was shorted during testing and a new PCB had to be designed. However, when designing the new PCB, there was a shortage of materials including the stepper motor driver, so the a4998 stepper motor driver was used instead and the new PCB was built around using this premade driver and attaching it to our circuit board. In the end, all these setbacks were overcome and the final project was successful and passed every aspect of our initial test plan.

Final Results

The end result of the capstone project resulted in two different methods of replication. One in which a user could upload an image file and run the python programs to replicate the writing. Another was an application that could allow the users to draw their signature or drawing in order to replicate what was drawn. Both of these traceable coordinates are logged in different files: coordinate.txt for the image file, and output.log for the application. This would place it in the A region for image processing. When these coordinates are sent, the microcontroller is able to successfully convert the coordinates into a replicated signature accurately and consistently, as shown below. Triple Tangent's successfully met the goals for the project that they established in the proposal. Appendix B contains pictures of the final product along with the original and replicated signature.

Grade	Image Processing	Embedded Programming	Overall Signature	Consistency
A	Able to convert image into traceable coordinates	Able to convert coordinates into a replicated signature	Automatic signature is similar to original signature	Reproduce signatures consistently
B	Able to somewhat convert image into traceable coordinates	Able to somewhat convert coordinates into replicated signature	Automatic signature is mostly similar to the original signature	Reproduce signatures fairly consistently
C	Mostly unable to convert image into traceable coordinates	Mostly unable to convert coordinates into replicated signature	Automatic signature is dissimilar but not completely off	Reproduce signatures rarely
D	Unable to convert image into traceable coordinates	Unable to convert coordinates into replicated signature	Automatic signature does not resemble the original signature.	Cannot reproduce signatures.

Table 1: Proposal Requirements Chart

However, there were a couple of additional goals that the team did not meet. The image processing algorithm does not result in an image that looks the same as the original image but rather very similar. The signature replication process also takes a nontrivial amount of time to replicate. It will take around a minute to replicate the original signature regardless of which method of input is used; however, the python application in which the user draws the signature is much faster than the image upload. There is also a 2 minute time period cooldown that is recommended for the machine as the electronics get very hot after an extended period of use. The additional goal of AES encryption was also not met as other than the standard encryption algorithms provided by the Bleak python module and in project zero of the TI RTOS Bluetooth server, there was not enough time to integrate our own custom security measures. This will be mentioned further in the future work section.

The end result has definitely taught the team a valuable lesson: anything that can go wrong will go wrong. Much more time should have been budgeted towards the beginning of the semester and the workload would not have been as end-heavy as it was.

Costs

A breakdown of the costs is shown in Appendix C. The entire cost of the project was totaled at 581.6$ for a single unit. While the team met the cost requirements as we were able to reuse the mechanical components, making this product from scratch would have exceeded the budget. The most expensive components were the mechanical components as a whole along with the microcontroller and PCB. When manufacturing a large number of units, the cost goes down by around 60$.

Future Work

Our current project is not very user-friendly, so our first major step would be to create a full application that is visually appealing and has all of our features easily accessible to the user. An application that combines both of our methods of replicating images that allow users to press a button to upload their image of choice or another button to go to the drawing replication canvas.

Also, the image processing algorithm is still very naive, and by using more advanced techniques such as Canny edge detection and more in-depth computer visualization methods, the image processing could be improved significantly. Another aspect that is lacking is our security measures. Right now we currently rely on the built-in security measures of the Bleak module and end-to-end encryption was not a priority when prototyping this device. If this device were to go to production, security would be a major factor of future work as much more research would have to go into security measures as signatures are highly sensitive information, and storing pictures of them should not be taken lightly.

References

[1] Texas Instrument, “MSP432E401Y SimpleLink™ Ethernet Microcontroller,” MSP432E401Y datasheet, Oct. 2017.

[2] “The history of the signature,” Legalesign, 19-Feb-2016. [Online]. Available: https://legalesign.com/blog/history-of-signatures/. [Accessed: 10-Sep-2021].

[3] L. L. Knutson, “Truman beach: The 33rd president at Key West,” WHHA (en-US). [Online]. Available: https://www.whitehousehistory.org/truman-beach-the-33rd-president-at-key-west. [Accessed: 10-Sep-2021].

[4] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, “Offline handwritten signature verification — literature review,” 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), 2017.

[5] “The ghostwriter: A workhorse that never sleeps,” Autopen Sales & Service | Automated Signature Technology, 2016. [Online]. Available: http://signaturemachine.com/ghostwriter/. [Accessed: 10-Sep-2021].

[6] “QMOT QSH4218 Manual.” [Online]. Available: https://www.trinamic.com/fileadmin/assets/Products/Motors_Documents/QSH4218_manual_Rev1.08.pdf. [Accessed: 17-Dec-2021].

[7] E. Chung, “Bioplastics: What they are and why they’re not a perfect solution | CBC

News,” CBC, Mar. 05, 2020. https://www.cbc.ca/news/technology/bioplastics-backgrounder1.5485009 (accessed Dec 17, 2021).

[8] J. Papiewski, “How long do processors last?,” Small Business - Chron.com, 21-Nov-2017. [Online]. Available: https://smallbusiness.chron.com/long-processors-last-71800.html. [Accessed: 20-Sep-2021].

[9] Rob, “What is the lifespan of a 3D printer?,” 3D Printscape, 06-Jul-2021. [Online]. Available: https://3dprintscape.com/what-is-the-lifespan-of-a-3d-printer/. [Accessed: 20-Sep-2021].

[10] “What are the factors that will affect the PCB lifespan and how to extend it: 2021 newest,” What are the Factors that will Affect the PCB Lifespan and How to Extend It | 2021 newest, 18-Mar-2020. [Online]. Available: https://www.pcbonline.com/blog/How_Is_The_Lifespan_of_PCB_Products_Determined_117.html. [Accessed: 20-Sep-2021].

[12] “United States Power Adapter - electrical outlets & plugs,” World, 2021. [Online]. Available: https://world-power-plugs.com/united-states. [Accessed: 20-Sep-2021].

[13] Webmaster, “Embedded C coding standard,” Barr Group Software Experts, 26-May-2016. [Online]. Available: https://barrgroup.com/embedded-systems/books/embedded-c-coding-standard. [Accessed: 17-Dec-2021].

[14] “About NEMA standards,” NEMA. [Online]. Available: https://www.nema.org/standards/about-standards. [Accessed: 17-Dec-2021].

[15] “Association connecting Electronics Industries - IPC.” [Online]. Available: https://www.ipc.org/TOC/IPC-2221A.pdf. [Accessed: 17-Dec-2021].

[16] “IEEE 802.15.1-2002 - IEEE standard for telecommunications and information exchange between systems - LAN/man - specific requirements - part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks (wpans),” IEEE SA - The IEEE Standards Association - Home. [Online]. Available: https://standards.ieee.org/standard/802_15_1-2002.html. [Accessed: 17-Dec-2021].

[17] “PEP 8 -- style guide for python code,” Python.org. [Online]. Available: https://www.python.org/dev/peps/pep-0008/. [Accessed: 17-Dec-2021].

[18] “Electronics Industries Generic Requirements for ... - IPC.” [Online]. Available: https://www.ipc.org/TOC/IPC-7351.pdf. [Accessed: 17-Dec-2021].

[19] “STL File Format for 3D Printing - Simply Explained,” All3DP, 13-Jun-2017. [Online]. Available: https://all3dp.com/what-is-stl-file-format-extension-3d-printing/. [Accessed: 15- Dec-2017].

[20] Electronics Notes. “USB Universal Serial Bus standards.” electronics-notes.com.

https://www.electronics-notes.com/articles/connectivity/usb-universal-serialbus/standards.php (accessed Dec. 16, 2019).

[21] “Transistor Outline (TO) Package Types,” Transistor outline (TO) package types. [Online]. Available: https://eesemi.com/to-types.htm. [Accessed: 17-Dec-2021].

[22] “FIPS 197, Advanced Encryption Standard (AES) - NIST.” [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf. [Accessed: 17-Dec-2021].

[23] “Autodesk empowers innovators everywhere to make the new possible,” Autodesk, 15-Nov-2021. [Online]. Available: https://www.autodesk.com/. [Accessed: 17-Dec-2021].

[24] “3D printers for Educators & Professionals,” MakerBot. [Online]. Available: https://www.makerbot.com/. [Accessed: 17-Dec-2021].

[25] National Instruments. “Multisim download.” ni.com. https://www.ni.com/enus/support/downloads/software-products/download.multisim.html#312060 (accessed Dec. 16, 2019).

[26] National Instruments. “Ultiboard download.” ni.com. https://www.ni.com/en-us/support

/downloads/software-products/download.ultiboard.html#311427 (accessed Dec. 16, 2019).

[27] Advanced Circuits. “2 & 4 layer PCB special pricing options.” 4pcb.com. https://www

.4pcb.com/pcb-prototype-2-4-layer-boards-specials.html (accessed Dec. 16, 2019).

[28] “What is VirtualBench?,” NI. [Online]. Available: https://www.ni.com/en-us/shop/electronic-test-instrumentation/virtualbench/what-is-virtualbench.html. [Accessed: 17-Dec-2021].

[29] Texas Instruments. “Code Composer Studio downloads.” ti.com. http://software-dl.ti

.com/ccs/esd/documents/ccs_downloads.html (accessed Dec. 16, 2019).

[30] “Welcome to Python.org,” Python.org. [Online]. Available: https://www.python.org/. [Accessed: 17-Dec-2021].

[31] “PyCharm: The python IDE for professional developers by jetbrains,” JetBrains. [Online]. Available: https://www.jetbrains.com/pycharm/. [Accessed: 17-Dec-2021].

[32] OpenCV, 08-Dec-2021. [Online]. Available: https://opencv.org/. [Accessed: 17-Dec-2021].

[33] NumPy. [Online]. Available: https://numpy.org/. [Accessed: 17-Dec-2021].

[34] Bleak. [Online]. Available: https://bleak.readthedocs.io/en/latest/. [Accessed: 17-Dec-2021].

[35] “US10108869B2 - method and device for reproducing content,” Google Patents. [Online]. Available: https://patents.google.com/patent/US10108869B2/en?q=replication%2Bof%2Bpencil%2Bstrokes&oq=replication%2Bof%2Bpencil%2Bstrokes. [Accessed: 17-Dec-2021].

[36] “US8700905B2 - method and device for electronically capturing a handwritten signature using embedding technique,” Google Patents. [Online]. Available: https://patents.google.com/patent/US8700905B2/en?q=autosignature&oq=autosignature. [Accessed: 17-Dec-2021].

[37] “The ghostwriter: A workhorse that never sleeps,” Autopen Sales & Service | Automated Signature Technology. [Online]. Available: http://signaturemachine.com/ghostwriter/. [Accessed: 17-Dec-2021].

[38] L. S. Ennes, “Schukuratsu/python-cv2-fast-thinning-algorithm: Implementation of a fast thinning algorithm using morphology,” GitHub, 16-May-2018. [Online]. Available: https://github.com/Schukuratsu/Python-cv2-fast-thinning-algorithm. [Accessed: 10-Sep-2021].

[39] “Denoising,” OpenCV. [Online]. Available: https://docs.opencv.org/4.x/d1/d79/group__photo__denoise.html. [Accessed: 17-Dec-2021].

[40] Welcome to Kivy - Kivy 2.0.0 documentation. [Online]. Available: https://kivy.org/doc/stable/. [Accessed: 17-Dec-2021].

[41] “Microsoft,” Microsoft Support. [Online]. Available: https://support.microsoft.com/en-us/surface/surface-pro-4-features-e3a8ac71-4cee-9107-23bc-7bd08d47cf75. [Accessed: 17-Dec-2021].

[42] “Boostxl-CC2650MA,” BOOSTXL-CC2650MA Evaluation board | TI.com. [Online]. Available: https://www.ti.com/tool/BOOSTXL-CC2650MA. [Accessed: 17-Dec-2021].

[43] “project_zero,” Ti devtools. [Online]. Available: https://dev.ti.com/tirex/explore/node?node=AEkrKeACNPV6UYYXi34bZA__kmPly-e__LATEST. [Accessed: 17-Dec-2021].

[44] IEvangelist, “How to: Implement a producer-consumer dataflow pattern,” Microsoft Docs. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/standard/parallel-programming/how-to-implement-a-producer-consumer-dataflow-pattern. [Accessed: 17-Dec-2021].

[45] “Stepper motor in 3D printers: Whiteclouds,” WhiteClouds, 07-Oct-2021. [Online]. Available: https://www.whiteclouds.com/3dpedia/stepper-motor/#:~:text=Stepper%20motors%20are%20an%20important%20part%20of%203D%20printers.&text=Stepper%20motors%20are%20unique%20in,in%20a%20variety%20of%20sizes. [Accessed: 17-Dec-2021].

[46] Rob, “Why do 3D printers use Stepper Motors? (we've got the answer),” 3D Printscape, 22-Jul-2021. [Online]. Available: https://3dprintscape.com/why-do-3d-printers-use-stepper-motors/. [Accessed: 17-Dec-2021].

[47] “How a Stepper Motor Works,” HowToMechatronics, 07-Oct-2021. [Online]. Available: https://howtomechatronics.com/how-it-works/electrical-engineering/stepper-motor/. [Accessed: 17-Dec-2021].

[48] “QMOT QSH4218 Manual.” [Online]. Available: https://www.trinamic.com/fileadmin/assets/Products/Motors_Documents/QSH4218_manual_Rev1.08.pdf. [Accessed: 17-Dec-2021].

[49] “DMOS microstepping driver with translator and ... - pololu.” [Online]. Available: https://www.pololu.com/file/0J450/A4988.pdf. [Accessed: 17-Dec-2021].

[50] “Pololu - A4988 Stepper Motor Driver Carrier,” Pololu Robotics & Electronics. [Online]. Available: https://www.pololu.com/product/1182. [Accessed: 17-Dec-2021].

[51] “Features s applications – coupling – filtering – de - coupling.” [Online]. Available: https://www.cde.com/resources/catalogs/KXM.pdf. [Accessed: 17-Dec-2021].

[52] “Advanced circuits,” Printed Circuit Board Trace Width Tool | Advanced Circuits. [Online]. Available: https://www.4pcb.com/trace-width-calculator.html. [Accessed: 17-Dec-2021].

[53] “The basics of solenoids and electric motors,” Singapore Travel Guide. [Online]. Available: https://www.streetdirectory.com/travel_guide/134734/technology/the_basics_of_solenoids_and_electric_motors.html. [Accessed: 17-Dec-2021].

[54] “Model L-90 DC Solenoid Open Frame, Pull Type.” [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Avago%20PDFs/ADNS-3050.pdf%3E. [Accessed: 17-Dec-2021].

[55] “LM62460, LM61480, and LM61495 pin-compatible 6-A/8-A/10-A ...” [Online]. Available: https://www.ti.com/lit/ds/symlink/lm61495.pdf?ts=1634225117018&ref_url=https%253A%252F%252Fwww.ti.com%252Fpower-management%252Fnon-isolated-dc-dc-switching-regulators%252Fstep-down-buck%252Fbuck-converter-integrated-switch%252Foverview.html. [Accessed: 17-Dec-2021].

[56] H. Powell, “Basic BJT Relationships.” [Online]. Available: https://collab.its.virginia.edu/access/content/group/0b23060e-0564-4e66-b135-82bac6a38fe9/Reading%20Materials/Basic%20BJT%20Relationships-Rev6.pdf.

[57] “The flyback diode: Voltage problems and switching solutions,” Cadence, 26-Mar-2020. [Online]. Available: https://resources.pcb.cadence.com/blog/2019-the-flyback-diode-voltage-problems-and-switching-solutions#:~:text=To%20prevent%20the%20reverse%20polarity,it%20in%20the%20opposite%20direction. [Accessed: 17-Dec-2021].

[58] "MCV 1,5/2-G-3,81," Phoenix Contact. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Avago%20PDFs/ADNS-3050.pdf%3E. [Accessed: 17-Dec-2021].

[59] "uClamp3301H," SEMTECH. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/44000000MDWf/s0KJxyOHQyYOcvBXVBK9X3X8Cxj1gNItZ2vd2k1S0Wo. [Accessed: 17-Dec-2021].

[60] "AC-DC wall adapter." [Online]. Available: https://globtek.secure.force.com/spec/?id=01t0c000008OwRv. [Accessed: 17-Dec-2021].

[61] "R-78 -0 - RECOM Power." [Online]. Available: https://recom-power.com/pdf/Innoline/R-78-0.5.pdf. [Accessed: 17-Dec-2021].

[62] "Regulations - RECOM power." [Online]. Available: https://recom-power.com/pdf/Innoline/R-78C-1.0.pdf. [Accessed: 17-Dec-2021].

[63] "DRV8436 stepper driver with integrated current ... - ti.com." [Online]. Available: https://www.ti.com/lit/ds/symlink/drv8436.pdf. [Accessed: 17-Dec-2021].

[64] "PWM LOW-SIDE DRIVER (1.5A and 3A) for Solenoids, Coils, Valves, Heaters, and Lamps." [Online]. Available: https://www.ti.com/lit/ds/symlink/drv103.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1639715196298&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Fdrv103.

[65] J. Tang, "Motor sizing basics part 1: How to calculate load torque," [Online]. Available: https://blog.orientalmotor.com/motor-sizing-basics-part-1-load-torque. [Accessed: 17-Dec-2021].

Appendix

Appendix A: Block Diagram

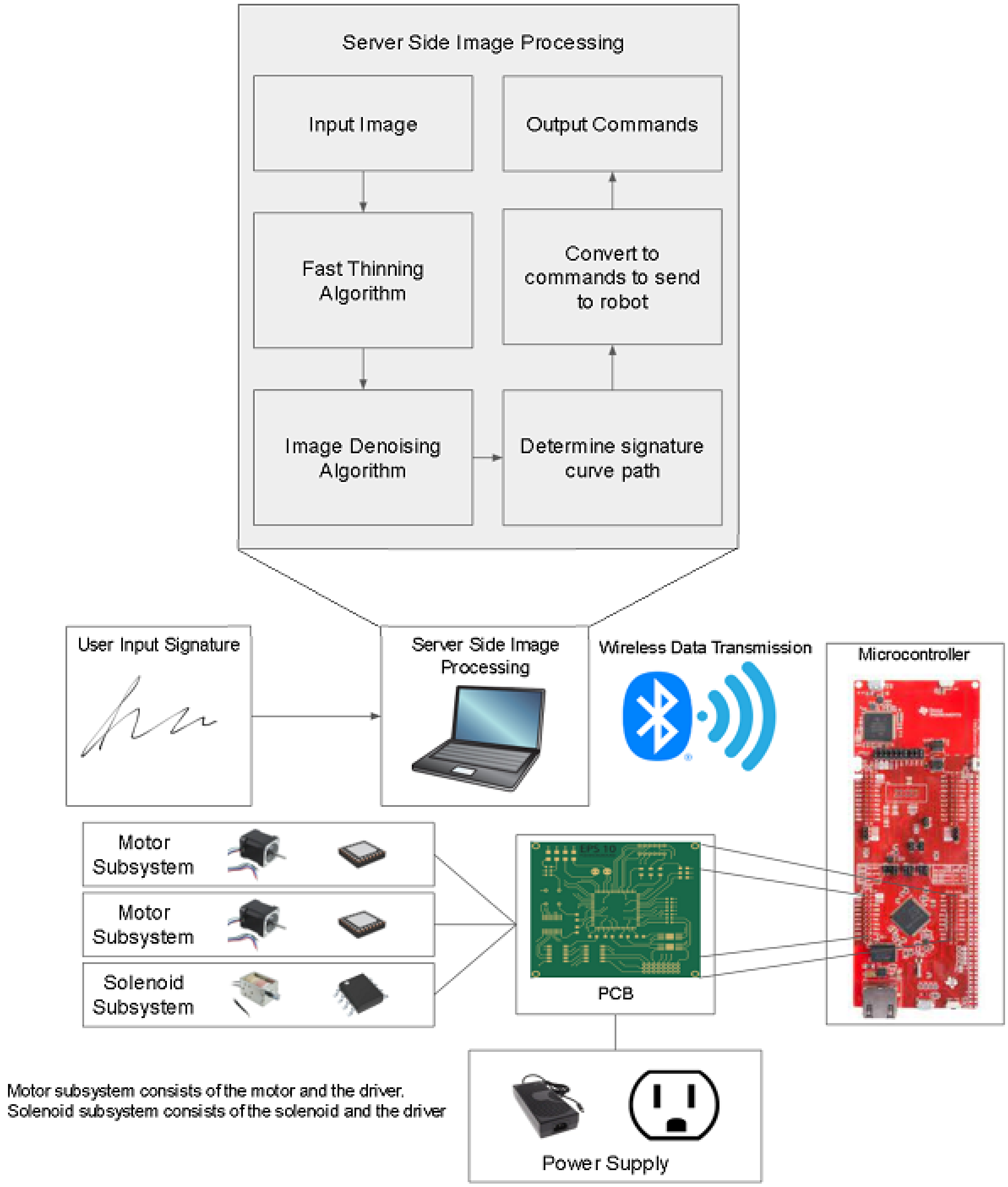


Figure 37: Project Block Diagram

Appendix B: Final Product Pictures

The pictures below contain the entire device as a whole. The other two pictures contain first the original signature side by side to the replicated signature done by the machine.

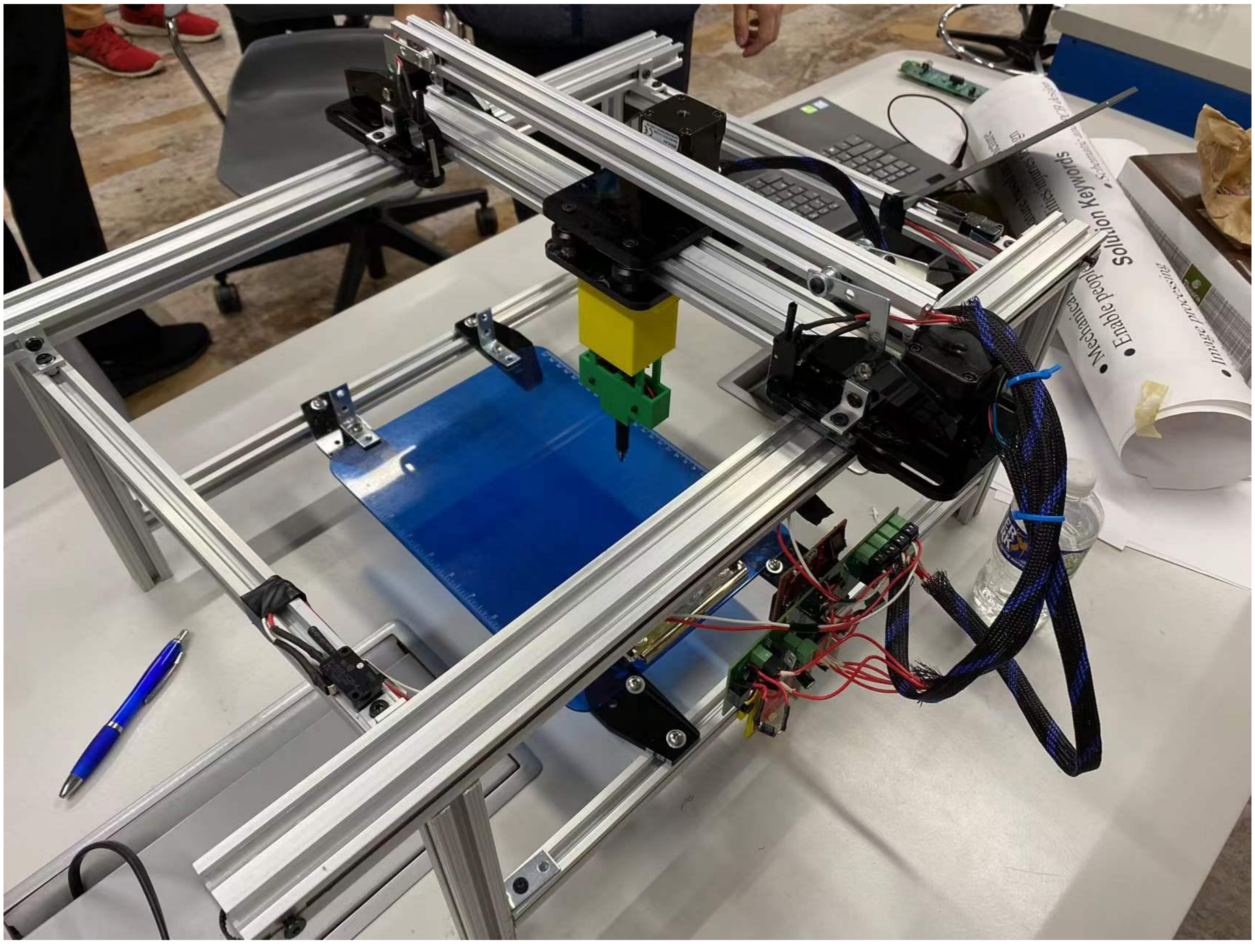


Figure 38: Project Final Product

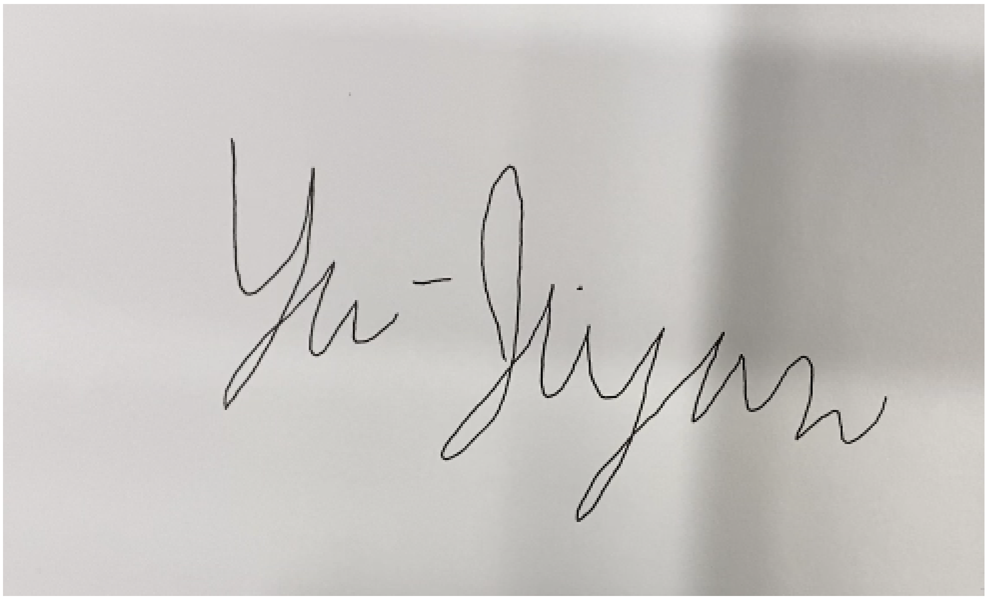


Figure 39: Replicated Signature

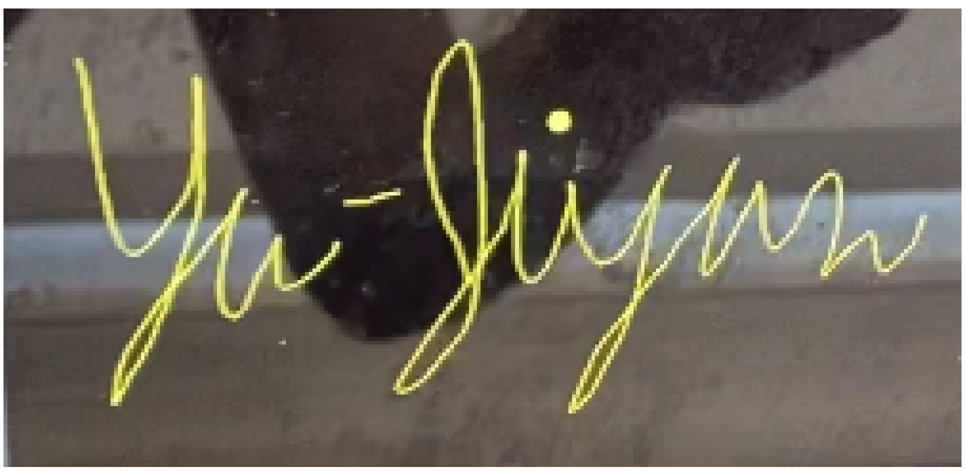


Figure 40: Original Signature

Appendix C: Costs Chart

Product	Cost for 1 unit	Cost per unit for 10,000 units	Qty
CC2650 BLE BOOSTERPACK BOARD	34.8	34.8	1
MSP-EXP432E401Y	47.99	47.99	1
Stepper Motor	47.3	43.38	2
Solenoid Motor	34.87	26.15	1
Power Adapter	62.21	51.71	1
Power Jack	1.59	0.82	1
Switch	4.27	2.34	4
Schottky Diode TH	0.57	0.29	2
DC-DC 3.3V Converter	7.83	6.26	1
DC-DC 9V Converter	10.12	8.27	1
Schottky Diode SMH	0.5	0.19	1
1K Resistor SMH	0.1	0.00659	9
0.01uF Ceramic Capacitor SMH	0.11	0.0222	4
BJT	3.86	2.4	2
A4988 Stepper Motor Driver Carrier	5.47	5.47	2
20 Pin Connector	1.26	0.67	2
8 Pin Convertor	0.54	0.22	4
2 Pos Terminal Block Header	0.86	0.55	4
2 Pos Terminal Block	2.48	1.73	4

Plug			
4 Pos Terminal Block Header	1.79	1.15	2
4 Pos Terminal Block Plug	4.98	3.36	2
100uF Electrolytic Capacitor	0.47	0.12	2
1k Resistor TH	0.1	0.00662	1
1x2 Header Pin	0.06	0.029	4
Acro Axis Plate Bundle (Main, Left, Right)	29.99	29.99	1
T-Nut (single)	0.299	0.299	40
T-Nut (double)	0.69	0.69	6
32 Thread Screws (10mm)	0.109	0.109	36
32 Thread Screws (20mm)	0.129	0.129	10
32 Thread Screws (40mm)	0.149	0.149	12
Belt/Pinion set	14.99	14.99	1
L-Brackets (single)	1.29	1.29	10
L-Brackets (double)	1.49	1.49	4
Corner Brackets	2.99	2.99	2
V-Slot (20x40x2000mm)	27.99	27.99	1
V-Slot 3(20x20x1500mm)	15.99	15.99	1
V-Slot (20x20x1000mm)	10.69	10.69	1

T-Slot (1x1x4ft)	20.23	20.23	1
Washer	0.19	0.19	18
Spacer	0.249	0.249	20
Acro Mini Plate	9.49	9.49	2
Clipboard	2.75	2.75	1
Solid V Wheel	5.19	5.19	8
Delrin V Wheel	4.99	4.99	2
Total	**581.6**	**525.51**	

Table 2: Costs Breakdown Chart

IS SAAS THE FUTURE? A CASE STUDY OF SHOPIFY'S BUSINESS MODEL AND SOCIAL CONSTRUCTION

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Zichao Hu

Spring 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature Zichao Hu ______________________ Date 05/03/2022

Approved ______________________ Date __________

ABSTRACT

Software as a service (SaaS) is a novel software distribution model to license and deliver applications to end-users. It has gained much popularity today as many leading companies have adopted this model into their business models. People become increasingly interested in SaaS and start pondering if SaaS is the future. This work will investigate the question by performing a case study on Shopify's business model. It will employ the Social Construction of Technology (SCOT) as the primary methodology. The goal of the paper is to study how different groups interpret the SaaS model and understand the role of SaaS in society.

IS SAAS THE FUTURE? A CASE STUDY OF SHOPIFY'S BUSINESS MODEL AND SOCIAL CONSTRUCTION

INTRODUCTION

With the rise of the internet, new business models have been established. One of the results of the digitization efforts has been the rise of software as a service (SaaS). SaaS is a novel software distribution model to license and deliver applications to end-users today (Chai, 2021). In this model, the software provider will host its ready-to-use applications on its cloud server so that the clients can simply log onto the host's website and immediately use all the applications without going through complicated software installment and setups. Some of the most well-known companies today, such as Salesforce, Shopify, and Google, offer products in the Saas model and these products are the main driving venues for the company's growth (Patrizio, 2020).

With increasingly more companies embracing the progress of digitization as part of their long-term business strategy, it becomes a hot topic in the capital world whether SaaS is the future (Davis, 2022). This work is going to perform extensive research on this question. Particularly, it is going to perform a case study of the business model and the social impacts of Shopify, a leading and rapidly growing SaaS company in the eCommerce industry today. The main research method will be the Social Construction of Technology (SCOT) framework proposed by Trevor Pinch and Wiebe Bijker (Bijker & Pinch, 1984) as the primary approach. The Social Construction of Technology (SCOT) argues that technology does not determine human action but instead human action shapes technology (Bijker & Pinch, 1984). Following this method, this research will first investigate Shopify's business model and analyze how the company incorporates SaaS. Second, the work will follow up on the business model of Shopify and examine how different social groups participate in Shopify's SaaS model. Particularly, this work will focus on six different social groups: the business (Shopify), clients (merchants), developers (third-party developers), growth investors, ordinary customers, and outside non-users, as these groups have different stakes in the evolution of SaaS technology and they represent a significant portion of the society. Furthermore, this work will look into the conflicts of interests as a result of different interpretations. The goal of this research is to gain a deeper understanding of the SaaS model.

SHOPIFY'S BUSINESS MODEL

Shopify is a cloud-based platform that unifies all of a client's commerce into a single central command center. Merchants on Shopify are able to "customize their online store and sell in multiple places, including web, mobile, social media, online marketplaces, brick-and-mortar locations, and pop-up shops" (Voidonicolas, 2021). For example, Shopify's core product offers a selection of clean templates to get the website up and running and it includes a variety of essential functionalities such as integrated payment processing, search engine optimization (SEO) and marketing tools, and customer engagement tools. Additionally, Shopify also provides additional products and services to further customize the needs of the clients. There are thousands of third-party developed apps in Shopify's App Store to empower clients to grow their business. All of these solutions can significantly reduce the overhead time and cost of the development and maintenance of an eCommerce platform for Shopify's clients.

Shopify has two major revenue streams: subscription solutions and merchant solutions (Shopify Inc., 2022). For the subscription solutions, clients need to pay a monthly subscription fee to use the Shopify platform in the first place. Depending on the level of sophistication the client needs, Shopify offers various tiers of subscription plans to fulfill the needs. Furthermore, if the website still needs more customizations, clients can also choose to purchase additional services to upgrade the website. Shopify owns some of the additional services, but most of the products are developed by a third party. This feature is very much similar to Apple's App Store, in which the company provides a platform for developers to make apps and charges a percentage of the revenue the developers earned from the app. In contrast to Apple's 30% commission fee (Adorno, 2021), Shopify only charges 15% and exempts any fees for smaller developers with less than one million dollars in revenues. For merchant solutions, Shopify principally generates revenues from payment processing fees and currency conversion fees from Shopify Payments. In another word, Shopify operates similarly to a credit card issuer where it provides easy payment processing tools for both the merchants and their customers, and in return shares a percentage of the profits from the merchants.

By analyzing these two major revenue streams, Shopify's business logic becomes apparent. As shown in Figure 1, Shopify has its own business ecosystem. In the beginning, Shopify offered its core services, such as webpage management, payment processing, and advertisement tools, to the merchants so that the merchants can quickly build a functioning website to sell their products. The subscription fees collected by Shopify allow it to provide more support for third-party developers by lowering the commission fees. As a result, developers may find it more lucrative to build applications on Shopify's platform, so this could greatly increase the utility of the Shopify Store and address problems that a merchant could encounter in their eCommerce business. With better support, merchants can grow their business bigger and faster and generate more revenue. Shopify also gains from the growth of its merchant clients because it shares a percentage of the profits with the merchants via the payment processing services. As Shopify makes more profits from the business, it reinvests these profits back to grow its business and attract more merchants and third-party developers to join its ecosystem. Ultimately, this leads to a virtuous cycle. Shopify is able to build its economic moat from this business model and quickly surpass potential competitors.

Figure 1. Shopify: The E-commerce on-ramp. This figure, excerpted from Shopify's business model report, illustrates the ecosystem of Shopify and demonstrates its business model. (Danco, 2021)

THE SOCIAL CONSTRUCTION OF SAAS

The Social Construction of Technology (SCOT) argues that technology does not determine human action but instead human action shapes technology (Bijker & Pinch, 1984). To study how human action shapes technology, it is important to first understand what groups are relevant. According to Pinch and Bijker (Bijker & Pinch, 1984), a relevant social group consists of "all members of a certain social group [who] share the same set of meanings, attached to a specific artifact. The relevant social groups interpret artifacts as having some sort of meaning (positive or negative) that can secure one innovation pathway over another or halt innovation entirely." One important concept in SCOT is interpretative flexibility, which means that each technological artifact has different meanings and interpretations for various groups. During the period of interpretive flexibility, SCOT portrays "alternative designs as distinct artifacts that might appeal to more and more groups." Hence, it is important to look at how the criteria of SaaS are defined and what groups and stakeholders participate in defining it. This work mainly focuses on six social groups that relate to SaaS: Shopify Inc, the merchants, the third-party developers, the growth investors, the ordinary customers, and the outside non-users. The following sections will invoke the SCOT model to investigate how each of these relevant groups has different interpretations of SaaS.

SAAS AND SHOPIFY

Shopify is a typical SaaS company that sells cloud-based software at a monthly subscription fee. The business model of Shopify perfectly revolves around the SaaS model. Within Shopify, each division can have its definition of how the SaaS.

First, Shopify's development team is mostly concerned with developing and managing products. The development team aims to find technology that can boost its efficiency and reduce the overhead of development. As a result, Shopify's development team interprets the SaaS model to be a lightweight framework to manage its platform efficiently (Zylo, 2022). Since SaaS solutions reside in a cloud environment, Shopify only has one platform to maintain on its cloud server. This gives Shopify centralized control over its product, for it is able to monitor the performance of its products and make incremental upgrades easily. In addition, there is a universal protocol for cloud-based services (Kerner, 2021), which abstracts away the specifics of clients' host machines and allows Shopify to serve all kinds of clients regardless of the clients' different machines. This contrasts with the traditional on-premises software, in which “an enterprise purchases a license or a copy of the software to use it” (Team Cleo, 2022). Since, on-premises software needs to be installed in the enterprise’s system environment, which tends to vary across different enterprises, the costs associated with managing and maintaining all the on-premises solutions entails can run exponentially higher than in a cloud computing environment. Hence it is much easier for Shopify’s development team to manage its products and services by using the SaaS model.

Second, Shopify’s management team, on the other hand, is responsible for the company’s long-term strategy and business operations. Growing the business and boosting revenue and sales are the major goals of the management team. Therefore, Shopify’s management team treats the SaaS model as a growth engine. As discussed above, cloud-based SaaS solutions can easily accommodate different clients with different backgrounds. This enables Shopify to scale its business rapidly and react fast to the clients’ ever-changing demands. This is supported by Shopify’s growth statistics (Thomas, 2022). As indicated in the report, Shopify had 4,2000 merchants in 2012. This number grew to 377,000 merchants in 2016, and finally, in 2020, Shopify had 1,749,000 merchants on its platform. As a result, Shopify has achieved an impressive 59% compounded growth rate in the number of merchants over the 8-year span. Moreover, Shopify’s management team also interprets the SaaS model as a novel solution to reduce its marginal cost. In economics, “the marginal cost of production is the change in total production cost that comes from making or producing one additional unit” (Tuovila, 2022). Different from a typical business model which roughly has a fixed amount of cost for producing an extra unit of product such as retail and hospitality business, the SaaS model allows Shopify to cut the marginal cost down to virtually zero as more and more people subscribe to its platform because the products are digital and the costs of more subscriptions do not depend on the number of subscriptions. This boosts the operational efficiency of the business and ultimately contributes to a high gross margin for Shopify of more than 50% (Shopify Inc., 2022).

SAAS AND MERCHANTS

Merchants are the clients of Shopify and they belong to a group of people that use Shopify's platform to build their eCommerce business. A common goal of this group is to build a successful eCommerce business so that they can make more profit. Hence, given what the SaaS model can offer, the merchants interpret the SaaS model in a few different ways.

First of all, the merchants interpret the SaaS model as a modern solution to perform proof of concept and deploy their eCommerce websites rapidly. Traditionally, if a merchant would like to open his/her eCommerce business, it is common to hire a professional web development team and spend months of development before the official launch date. This process is both time-consuming and expensive (Barrow, 2019). Moreover, it adds uncertainty because many things could happen during the development to cause the failure of the business. Today, with Shopify's simple templates and services, it is possible to deploy the website within a few hours of work and makes the whole process much less hassle-free. With the help of SaaS, the merchants are able to test out their ideas and get feedback faster. This allows the merchants to adapt their business to the customers' demands faster and thus is more likely to become successful.

Second, the merchants find it to have a lower barrier to entry to the eCommerce business because of the SaaS model employed by Shopify. Besides getting rid of hiring a professional team to develop the website, Shopify's SaaS model also simplifies every other aspect of running a business. For example, it provides essential services such as payment processing and marketing tools, which traditionally cannot be easily handled by the owner alone (Voidonicolas, 2021). Nowadays, because of the streamlined services provided by the SaaS model, it is less demanding on the clients' side to get the website running. Shopify has provided the necessary hardware and software for the clients already, such as powerful server machines and secure firewalls. It allows the clients to focus on their main eCommerce business without worrying about non-trivial technical issues.

SAAS AND THIRD-PARTY DEVELOPERS

A third-party developer is a developer not directly tied to the primary product that a consumer is using but the add-on product that can greatly enhance the functionality of the primary product. Today, as the SaaS model becomes widely adopted, there is a community of third developers who are willing to contribute to the development in exchange for monetary benefits. As the community grows, third-party developers can interpret the SaaS model as a tool to start their careers. Because the SaaS model is hosted on the Internet which is an open space and easily accessible (Berger, 2011), this lowers the third-party developers' barrier to entry. It becomes easier to contribute to the companies that adopt the SaaS model because there are standardized APIs and protocols for developers to follow. As a result, this enables the community of third-party developers to grow and helps the developers to make a profit. For example, Shopify has extensive partnerships with its third-party developers (Shopify Dev, 2022). According to Shopify's report (Shopify Inc., 2022), third-party developers made 12.5 billion dollars in 2020 from Shopify's Store. This is nearly four times as much as Shopify's revenue from its Store.

SAAS AND GROWTH INVESTORS

SaaS is certainly not an alien term for growth investors. Growth Investors are the group of people who take a special interest in finding growth opportunities so that they can commit capital and expect to receive an above-average return (Segal, 2021). As the SaaS companies gain more prominence on Wall Street and demonstrate how fast they can grow, investors interpret the SaaS model to be an alias for a high growth opportunity and a promising future (Copeland, 2014). Certainly, this belief among the investors is not unwarranted. For example, in Shopify's earning report (Shopify Inc. 2021), "the total revenue for the full year 2021 was $4,611.9 million, a 57% increase over 2020. Within this, Subscription Solutions revenue grew 48% to $1,342.3 million, while Merchant Solutions revenue grew 62% to $3,269.5 million." This has greatly surpassed the average 15.8% year-over-year revenue growth rate of the S&P 500 in 2021 (Butters, 2021). Another important indicator of success in the eCommerce industry is Gross Merchant Value (GMV). GMV is a metric that measures an eCommerce business's total value of sales over a certain period of time (Alloy Automation, 2021). The GMV of Shopify for 2021 was $175.4 billion, an increase of 49% over 2020. The previous GMV growth was respectively 99% in 2020 and 47% in 2019, which has shown great promise in Shopify's growth. All of these statistics have placed Shopify as one of the most popular growth stocks on Wall Street. Consequently, investors have great expectations of the SaaS companies which leads them to be willing to invest their money in the companies' stock.

SAAS AND ORDINARY CUSTOMERS

The SaaS model lowers the difficulty of running an eCommerce business which allows more and more small businesses can get involved. As a result, ordinary customers that buy products from the merchants will have more options to choose from. Specifically, Shopify's merchants tend to sell a variety of non-standard products that big retail manufacturers don't usually produce. Consequently, ordinary customers can think of the SaaS model as a way to amplify their choices. Moreover, as more and more small businesses get involved in the eCommerce business, the market will become increasingly competitive. Thus, it is more likely that ordinary customers can find a better deal on the products.

SAAS AND OUTSIDE NON-USERS

Since society is tightly coupled, any advancement of new technology will have an impact on all people in society regardless of whether the person is a user or not. Hence, the last relevant group in this analysis will examine the group of non-users. An example of non-users could be construction workers, whose labor work cannot be simplified nor compromised by the SaaS model. From the example, it is possible to argue that the direct impact of the SaaS model on non-users can be limited because the SaaS model works on the digital internet and could have a difficult time influencing the real world. Yet, in reality, the SaaS model has a more significant impact. For the same example of construction workers, even though the SaaS model does not contribute to their work in a meaningful way, the workers' employer might be a client who uses the SaaS services. For example, Workday Inc offers human resource management services for employers to pay their employees and manage logistics (Essex, 2016). In this way, the construction workers could receive their wages through the online SaaS services instead of a paper check. As a result, the non-users group could be indirectly influenced by the SaaS model and they may interpret the technology as an advancement in society that changes their previous lifestyle.

CONFLICT AND CLOSURE OF ALTERNATIVE INTERPRETATIONS

As discussed in the above sections, the interpretative flexibility of SCOT demonstrates that each of the different groups has its own views on the SaaS models. Yet, the different interpretations often give rise to conflicts between criteria. It tends to be challenging to resolve these conflicts between relevant groups. Fortunately, according to Bijker and Pinch, the issues of interpretive flexibility could collapse through closures (Bijker & Pinch, 1984). In this section, this work will analyze how these alternative interpretations generate different problems to be solved and provide possible closures.

First of all, while Shopify's development team interprets the SaaS model to reduce the developmental overhead by having centralized control over its product, this inevitably abstracts away the ability of merchants to have full control of their eCommerce business. Without having full control over the eCommerce business, the merchants' goal of building a successful business could be compromised. For example, although Shopify offers marketing services to their clients, the options of the services are very limited. This limitation could hurt the business because the services may not be customized exactly to the business and could cause the business to lose potential customers. This could come to closure by incorporating more third-party developers into Shopify's ecosystem, which aligns with the third-party developers' interpretations. More customized functionality could serve the merchants better. The introduction of another relevant group (third-party developers) can help resolve the problem.

Second, since Shopify's management team interprets the SaaS model as the growth engine, this will cause Shopify to expand the services more rapidly to more merchants. The management team strives to have more people subscribe to Shopify's services by lowering the difficulty of running an eCommerce business. On the other hand, the lowered barrier to entry could cause over-competition among the merchants. Merchants will have a harder time creating a successful business, which contradicts their interpretation of the SaaS model. This conflict could be resolved by coming to a rhetorical closure, in which social groups see the problem as being solved since one design is defined as being “good enough” to adopt one form over another. If the SaaS model is not adopted, this totally defeats the interpretations of the merchants as they will have an even more difficult time running their business. In addition, the over-competition among merchants aligns with the ordinary customers’ interpretations because it could give customers a better deal and allow them to have more options to buy from. Hence, the overall utility of society is increased.

Third, since SaaS companies tend to grow rapidly and report excellent earnings, a huge amount of capital is poured into the companies’ stock and the market has given these companies an extremely high valuation. This could easily create bubbles in the stocks’ valuation, which can diminish the investment returns for the investors and contradict the investors’ interpretations. For example, the historical price-to-sales ratio (P/S ratio) of Shopify achieved over 40, which greatly surpassed the average P/S ratio of 1.55 for the S&P 500 (Robinhood Learn, 2021). In addition, SaaS company stocks tend to have a higher fluctuation. For example, at the beginning of 2022, Shopify’s stock price tanked by half due to the fear of rate hikes (Cox, 2022) and various socio-political black swan events (NPR Staff, 2022). A redefinition of the problem could address these conflicts since growth investors could adopt a different valuation model for the growth companies in contrast to the traditional valuation model (Nguyen, 2022). Certainly, for companies growing so fast (Shopify Inc. 2021), it is unreasonable to compare them with other value stocks such as Coca-Cola or Nike. Moreover, if examining the stocks over a long period of time, the short-term fluctuations tend to become negligible. Hence, growth investors are able to make a profit as long as the companies they pick are indeed great companies.

CONCLUSION

Shopify Inc is a typical SaaS company as its business model perfectly revolves around the SaaS model. The first part of this work focuses on investigating the specifics of Shopify's business model by taking a closer look at its earnings report and business prospectus. Particularly, the first part of the work finds various statistics to analyze how the company operates and what role the SaaS model plays in the company's products. The analysis summarizes a clear line of business logic. As shown in Figure 1, Shopify first collects subscription fees for its core products so that it can lower its commission fee to attract more third-party developers to contribute and enhance the functionality of Shopify's core products; then, with better functionalities, more merchants will join Shopify's community and use its services; finally, Shopify shares a percentage of the profit with the merchants to boost its earnings so that investors are happy to put more capital in the company to help it grow. This forms a virtuous circle and eventually helps Shopify build its economic moat to surpass potential competitors.

The second part of this work uses the SCOT methodology to gain a deeper understanding of the SaaS mode given the social context. It reconstructs different groups' interpretations, such as Shopify itself, merchants, third-party developers, growth investors, ordinary customers, and non-users, given their stakes in this technology. It shows that different groups' interpretations influence the advancement of the SaaS model, which in turn has impacts on all the people in society today. Subsequently, the work discusses how these interpretations give rise to conflicts. This work addresses the problems through the use of rhetorical closure and redefinition of the problem employed by the SCOT methodology. Lastly, it is possible to conclude that the SaaS model increases the utility of society and it is the product of the digitization era.

References

Adorno, J (2021, Sep 14). Apple can still charge its App Store 30% fee even after Epic ruling. *9to5Mac.* https://9to5mac.com/2021/09/14/apple-can-still-charge-its-app-store-30-fee-even-after-epic-ruling-analysts-say/

Allender, E (2019, Aug 14). The pros and cons of software as a service. *Insight.* https://www.insight.com/en_US/content-and-resources/2019/08142019-the-pros-and-cons-of-software-as-a-service.html

Alloy Automation (2021, Oct 26). What is Gross Merchandise Value (GMV) and How Do You Increase It? *Alloy.* https://www.yieldify.com/blog/gross-merchandise-value-gmv/#:~:text=Gross%20Merchandise%20Value%20(GMV)%20is,to%20as%20Gross%20Merchandise%20Volume.

Barrow, E (2019, May 22). How long does it take to build an app? *3SidedCube*. https://3sidedcube.com/how-long-does-it-take-to-build-an-app/

Berger, J (2011, Oct 14). Digital public space: turning a big idea into a big thing. *BBC Internet Blog.* https://www.bbc.co.uk/blogs/bbcinternet/2011/10/digital_public_space_idea.html

Bijker, W. E., & Pinch, T. J. (1984). The social construction of facts and artifacts. *Social Studies of Science, 14, 399–441*. https://doi.org/10.1177/030631284014003004

Butters, J (2021, Dec 17). S&P 500 CY 2021 earnings preview: record-high earnings and sales growth in CY 2021. *Facts Set*. https://insight.factset.com/sp-500-cy-2021-earnings-preview-record-high-earnings-and-sales-growth-in-cy-2021

Byrne, E(2020, Mar 13). The good SaaS times will end and challenges are coming. *Tech Crunch*. https://techcrunch.com/2020/03/13/the-good-saas-times-will-end-and-challenges-are-coming/

Chai, W (2021, Feb). Software as a service (SaaS). *Tech Target.* https://www.techtarget.com/searchcloudcomputing/definition/Software-as-a-Service

Clark, B (2021). Is freelancing worth it? Career Sidekick.

https://careersidekick.com/freelancing-pros-cons/

Copeland, M (2014, May 27). Mind the gap: why Wall Street believes in SaaS companies. *A16z.* https://a16z.com/2014/05/27/wall-street-believes-in-saas-companies/

Cox, J (2022, Mar 16). Federal Reserve approves first interest rate hike in more than three years, sees six more ahead. *CNBC*. https://www.cnbc.com/2022/03/16/federal-reserve-meeting.html

Danco, A (2021, Apr 05). Shopify: The E-commerce on-ramp. *Join Colossus.* https://www.joincolossus.com/episodes/85946038/danco-shopify-the-e-commerce-on-ramp?tab=transcript

Danco, A (2021, Apr 05). Shopify: The E-commerce on-ramp. [Figure 1]. *Join Colossus.*

https://www.joincolossus.com/episodes/85946038/danco-shopify-the-e-commerce-on-ramp?tab=transcript

Daniel, M (2017, Mar 22). Why is cybersecurity so hard? Harvard Business Review. https://hbr.org/2017/05/why-is-cybersecurity-so-hard

Davis, J (2022). SaaS is the future of software, and that future is now. *Deloitte*. https://www2.deloitte.com/global/en/pages/technology/articles/saas-is-the-future-of-software-and-that-future-is-now-cloud-adoption-software-as-a-service-erp-oracle-digital-tranSformation.html

Hovhannisyan, S (2020, Oct 28). What are the most successful SaaS pricing models and what you may be doing wrong. *Incredo*. https://www.incredo.co/blog/successful-saas-pricing-models/

Nguyen, J (2022, Feb 19). How to choose the best stock valuation method. *Investopedia*. https://www.investopedia.com/articles/fundamental-analysis/11/choosing-valuation-methods.asp

NPR Staff (2022, Mar 24). A month into the Russia-Ukraine war: what happened today. *NPR*. https://www.npr.org/2022/03/24/1088519053/a-month-into-the-russia-ukraine-war-what-happened-today-march-24

Kerner, S (2021, May). Internet protocol (IP). *Tech Target.* https://www.techtarget.com/searchunifiedcommunications/definition/Internet-Protocol

Patrizio, A (2020, Sep 02). Top 75 SaaS companies of 2022. *Datamation.* https://www.datamation.com/cloud/saas-companies/

Robinhood Learn (2021, Mar 12). What is the price-to-sales ratio? Robinhood Inc. https://learn.robinhood.com/articles/01lnclx5jYAuQ0rCihxnlz/what-is-the-price-to-sales-p-s-ratio/

Segal, T (2021, June 04) Growth investing. *Investopedia*. https://www.investopedia.com/terms/g/growthinvesting.asp

Shopify Dev (2022). Shopify API reference docs. *Shopify Inc.* https://shopify.dev/api

Shopify Inc (2022, Feb 16). FORM 40-F. *Shopify Inc.* https://s27.q4cdn.com/572064924/files/doc_financials/2021/q4/99a0d08d-4015-4a0a-a8fb-fe6ec66553c4.pdf

Shopify Inc. (2021, Feb 16). Shopify announces fourth-quarter and full-year 2021 financial results. Shopify Press Releases. https://news.shopify.com/shopify-announces-fourth-quarter-and-full-year-2021-financial-results#:~:text=Total%20revenue%20for%20the%20full,increase%20of%2047%25%20over%202020.

Stewart, M (2022, Feb 17). Analysts still optimistic as Shopify (SHOP) stock plummets. *Capital.com.* https://capital.com/analysts-still-optimistic-as-shopify-shop-stock-plummets

Essex, D (2016, Dec). Workday. *TechTarget*. https://www.techtarget.com/searchhrsoftware/definition/Workday

Team Cleo (2022). On premise vs. cloud: key differences, benefits and risks. *Cleo*. https://www.cleo.com/blog/knowledge-base-on-premise-vs-cloud

Thomas, G (2022, Mar). Shopify statistics 2022: updated facts, market share & more. *AquireConvert.* https://acquireconvert.com/shopify-statistics/

Tuovila, A (2022, Feb 17). Marginal cost of production. *Investopedia*. https://www.investopedia.com/terms/m/marginalcostofproduction.asp

Voidonicolas, R (2021, Mar 10). What is Shopify? *Shopify Blog.* https://www.shopify.com/blog/what-is-shopify

Zylo (2022). What is SaaS management? Zylo. https://zylo.com/blog/saas-management/

Signature Replication Machine

(Technical Report)

SaaS And Internet

(STS Research Paper)

A Thesis Prospectus Submitted to the

Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements of the Degree
Bachelor of Science, School of Engineering

Zichao Hu
Spring, 2022

Technical Project Team Members

Zichao Hu

Edward Lee

Yu-Jiyun Tao

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature Zichao Hu ______ Date 05/03/2022

Approved ______ Date ______

Approved ______ Date ______

Signing has always been a source of security and privacy for past centuries. For example, the Statute of Frauds Act, which was passed by the Parliament of England in 1677, developed the concept of using signatures signing to seal contracts and acknowledge the approval of given documents (Legalesign., 2016). Today, signing becomes even more prevalent and crucial for validating legal documents (Knight, 2018). However, it can be repetitive, especially if a job requires signing large amounts of paperwork. With the technological advancement in autonomous robots and control systems, implementing an automatic signing machine would boost the efficiency and focus of the users and mitigate the possibility of injuries such as repetitive strain injury (Knott, 2017). Furthermore, the machine can serve to sign papers for users who are not able to physically.

The technical research will take on the quest of designing and building an automatic signing machine to address the mundane signing tasks in our daily life. The objective of the project is to free people from repetitive paperwork. The users only need a photo of their signatures that they can take physically or sign electronically and upload to a laptop. The laptop will control the machine via Bluetooth and automatically replicate their signatures for as many copies as needed. Thus, the signature will look similar to that signed by hand.

Meanwhile, the machine proposed by the research project is only made possible with the rise of the internet age, as the machine exemplifies an example of the internet of things (IoT). Today, although we can see how the internet age has such a drastic impact on mundane tasks, few works have been done to understand the question of "why." Hence, the tightly coupled STS research ponders why the internet significantly impacts mundane tasks. The research will investigate how different fields are revolutionized in the internet age first. Then the research will attempt to find common ground between those different fields and draw possible reasons to answer why the internet is revolutionary. Finally, this STS research project will be presented as a scholarly article.

This work will be accomplished during the Fall 2021 and Spring 2021 semesters. The detailed timelines are depicted in Figure 1.

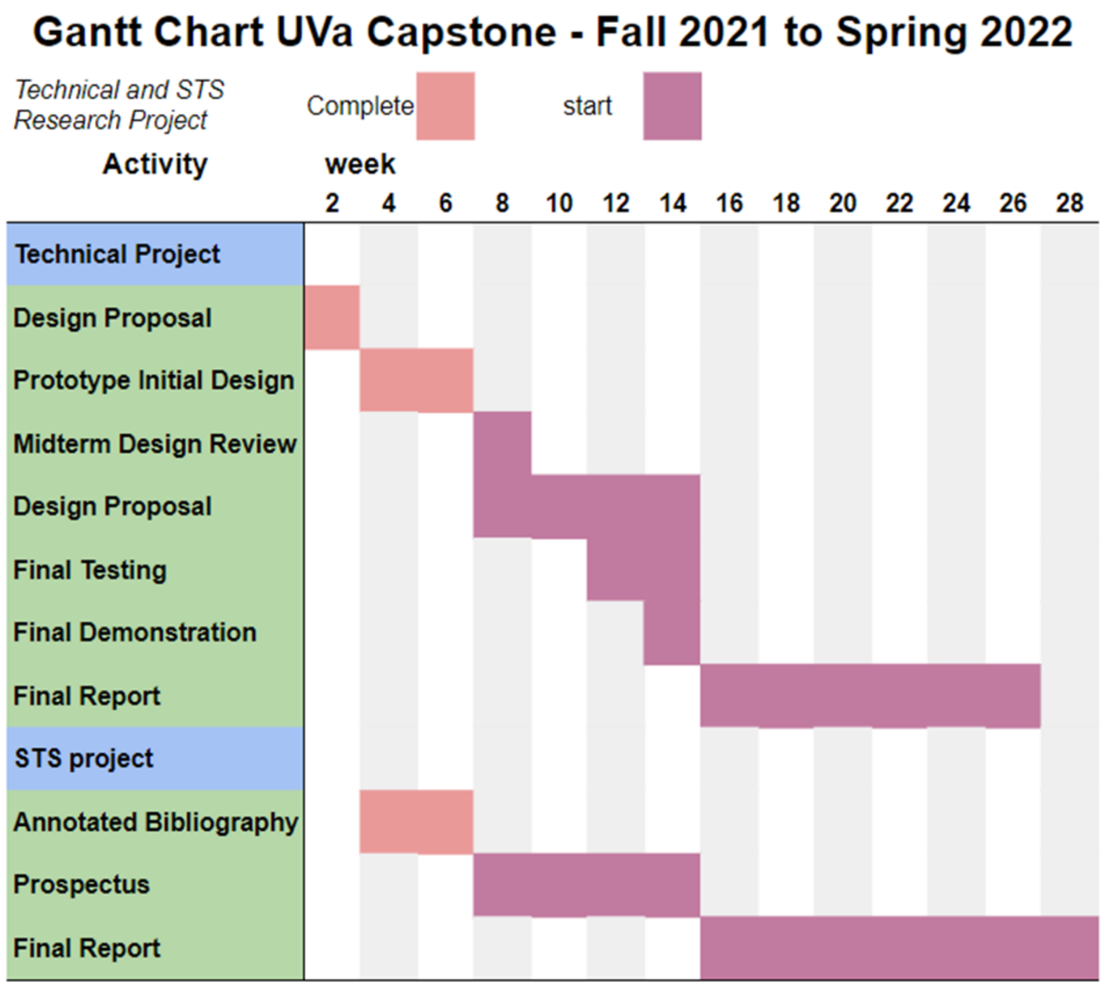


Figure 1: Gantt chart UVa automatic signing machine capstone. This figure visualizes the expected timeline for the major milestones achieved on the technical project and STS project. (Hu, 2021)

AUTOMATIC SIGNING MACHINE

Signing documents is essential and prevalent in everyday life because it “identifies the person who is a party to the contract and shows that the signing party has read the contents of the document, understands the contents, and consents to the stipulations of the contract” (Knight, 2018, p. 7). Signing documents involves many different tasks, such as signing a signature on a contract (Petrow, 2018), addressing a letter (Henry, 2019), and checking boxes in a form (Orem, 2021). However, it can be repetitive and onerous. For example, one might need to sign many agreements on his/her first day of work, such as an at-will agreement, non-compete agreement, and nondisclosure agreement (Guerin, 2021). Furthermore, some jobs require much paperwork, such as human resources, banking, and corporation operations. It can be overwhelming, for sometimes “[it] can feel like it is a never-ending process … when [one] gets home later, there will be more in the mailbox” (Kaysen, 2018, p. 5).

This research project aims to free humans from signing their documents physically by building an automatic signing machine. In this system, the user will upload a picture of their signature to a computer. The image processing algorithm on the computer will convert the image into commands that will be transferred over to the device, turning the commands into motion using stepper motors. These commands will allow the device to control the stepper motors to replicate the signature's image on paper. The overall flow is illustrated in Figure 2.

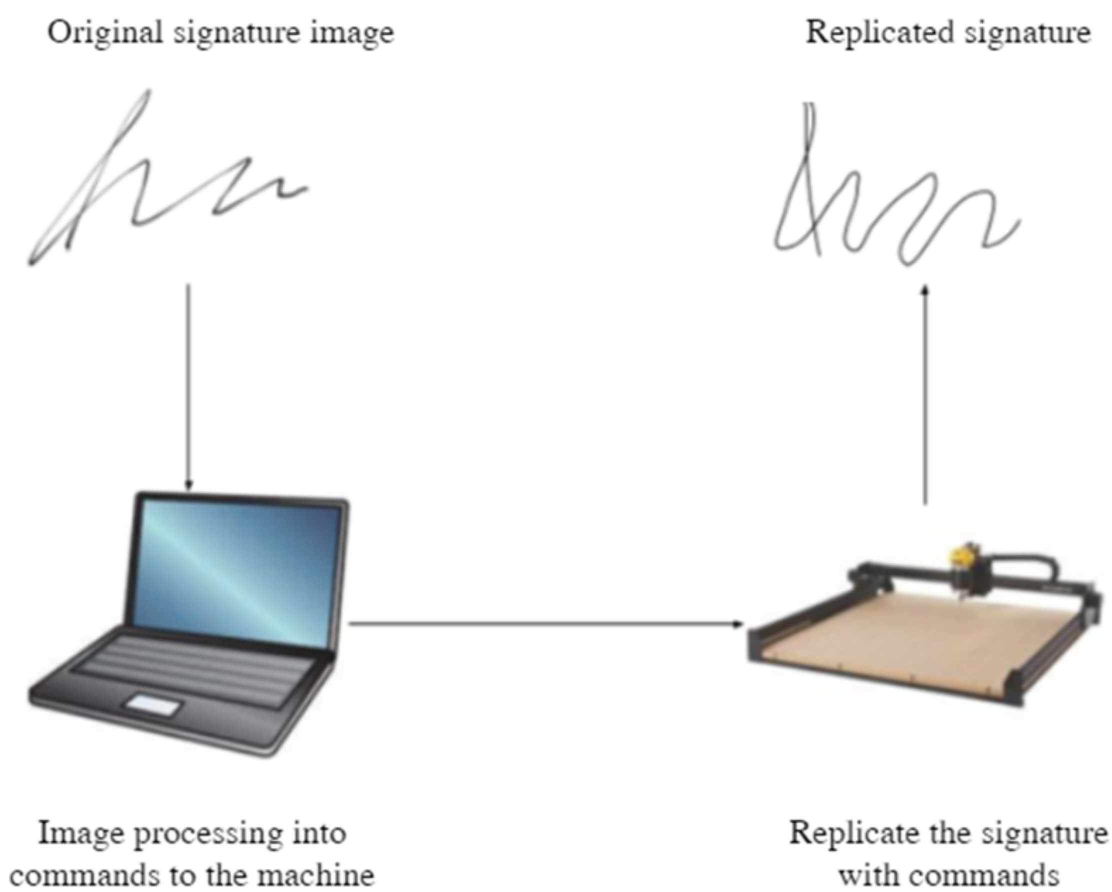


Figure 2. Automatic signature machine workflow. This figure shows the workflow of how to operate the designed system. (Hu, 2021)

There are three significant challenges presented in our research project. First, we need to design an image processing system on the server that can parse the image supplied by the user and convert it into commands to the signature machine. Second, we need to set up the communication via Bluetooth and ensure the safety of the data during transmission. Third, we need to design a PCB board that will interface with the microcontroller and the motors so that the server's commands can be executed correctly.

To address those challenges and meet the objective, I will work under the supervision of Electrical and Computer Engineering Professor Harry Powell. I will also work in a team with two students, Edward Lee and Yu-Jiyun Tao, who major in B.S. computer engineering. Together, we have experience in embedded programming, PCB designs, computer vision, image processing, and CAD designs. We will combine our knowledge and experience to deliver the project.

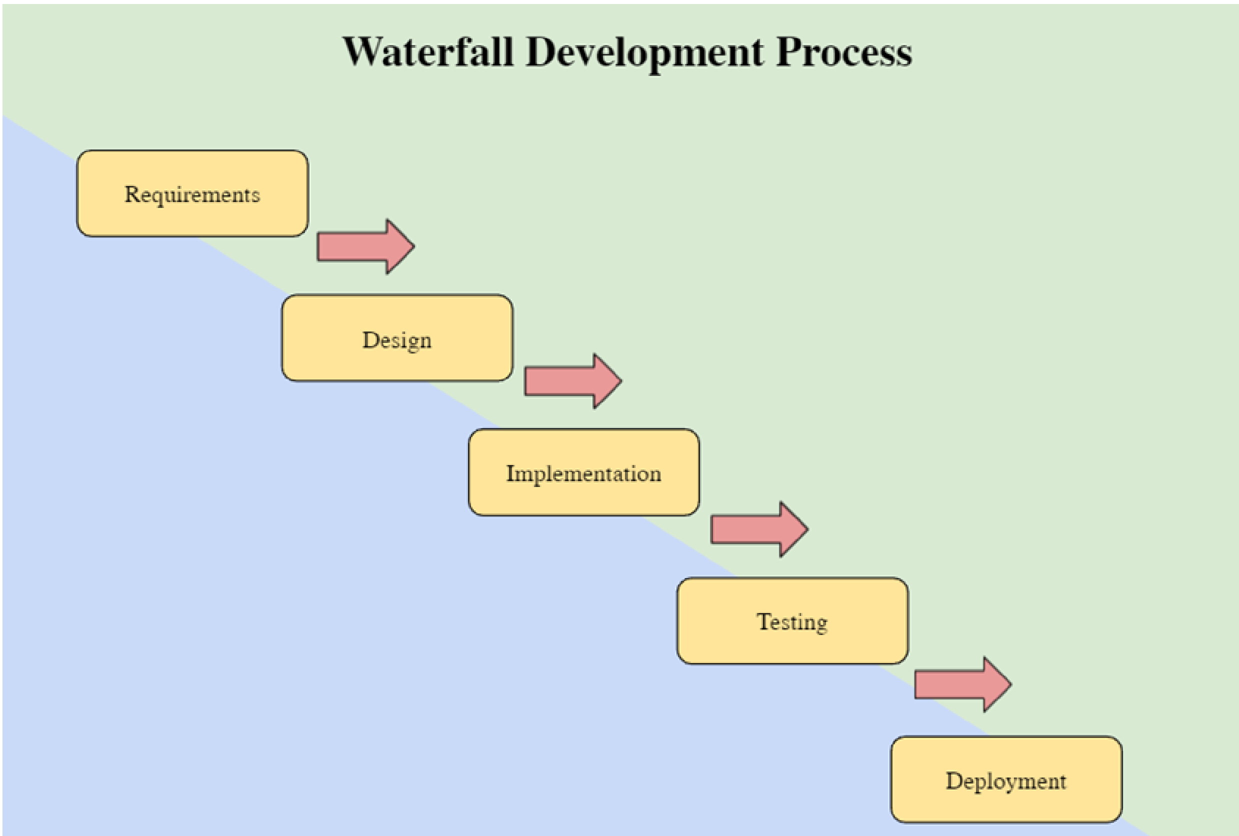


Figure 3. What is waterfall methodology in software development? This diagram demonstrates the overall process of the waterfall methodology. (Adapted by Hu (2021) from Sundar, 2018)

We will take the waterfall design approach that involves five consecutive steps: 1. Requirements, 2. Design, 3. Implementation, 4. Verification of Testing, 5. Deployment and Maintenance (Sundar, 2018). This is depicted in Figure 3. We chose the waterfall method because we need to design a PCB and a physical machine. Buying and building the parts will take quite a long time. Hence, it is reasonable for us to plan out initially and execute the plan accordingly afterward carefully.

National Instrument sponsors the research. We are provided with designing tools such as NI's Multisim and Ultiboard to design PCB. We are also provided with a workspace at the NI Lab inside the Thornton Hall. To complete the project successfully, we will also use OpenCV library, Digikey/Mouser, AutoDesk software, TI's SDK, and Code Composer IDE. In addition, we will consult the datasheets provided to us by the Texas Instrument (Texas Instrument, 2018). The expected outcome of the project is a fully functional automatic signature machine. The project will be documented in a technical report.

SaaS And Internet

Although the internet seems to be a pervasive thing today, it has only been about thirty years since its invention of the internet for public use (Grossman, 2018). If we compare human history to the ocean, then the history of the internet is only a drop of water in that ocean, as shown in Figure 4. Nevertheless, the arrival of the internet has had a profound impact on human society, and it is reasonable to agree that human civilization is welcoming a new Industrial Revolution: the era of the internet (Schwab, 2016). The era of the internet, or the internet age, is defined as "a time when large amounts of information are widely available to many people, largely through computer technology" (Thomas, 2019, p. 3). In this time, society undergoes enormous changes as we can see the emergence of the new and the abandonment of the old (Schwab, 2016).

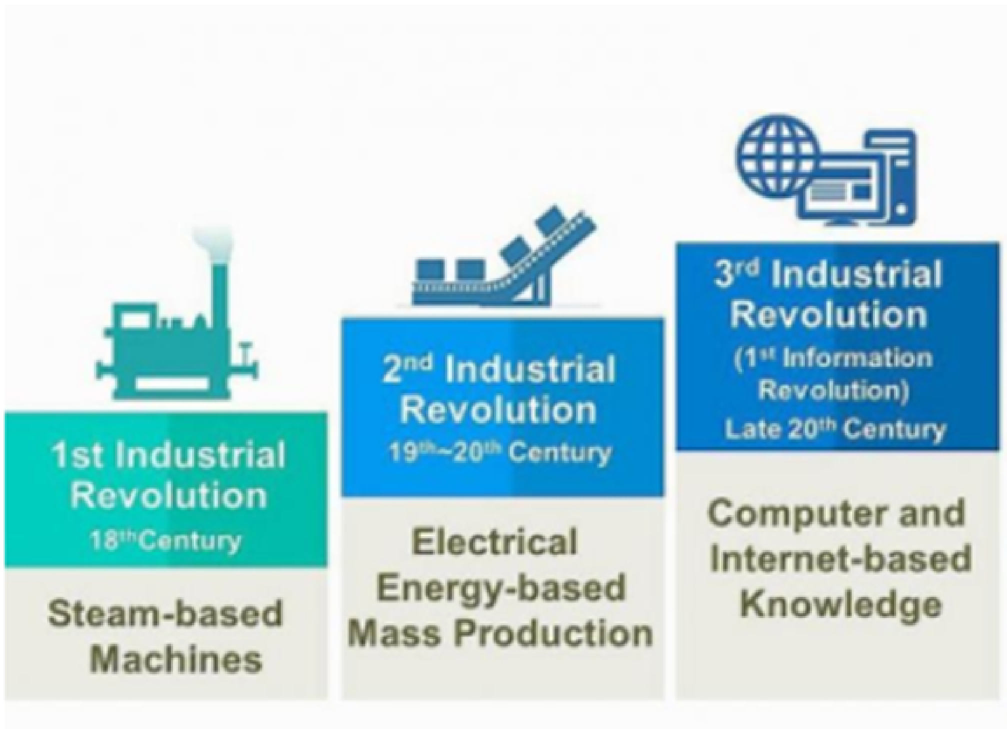


Figure 4. Different periods of the industrial revolution. This diagram shows the progress of human civilization and demonstrates the short history of computers in human history (Kearns, 2019).

IMPACT OF THE INTERNET ON DAILY TASKS

Although the internet is a “newcomer,” people have seen drastic changes in many tasks, and many works have studied the impacts (Pew Research Center, 2004). According to the Pew Research Center, “92% of internet users say the Internet is a good place to go for getting everyday information” (Pew Research Center, 2004, p. 7). The statistics show that increasingly more people turn to the internet today, as illustrated in Figure 5. For example, people tend to purchase tickets online in advance to a sports game or a movie rather than buy tickets at the selling station. In addition, people rely heavily today on digital maps on their phones when driving instead of using a physical map.

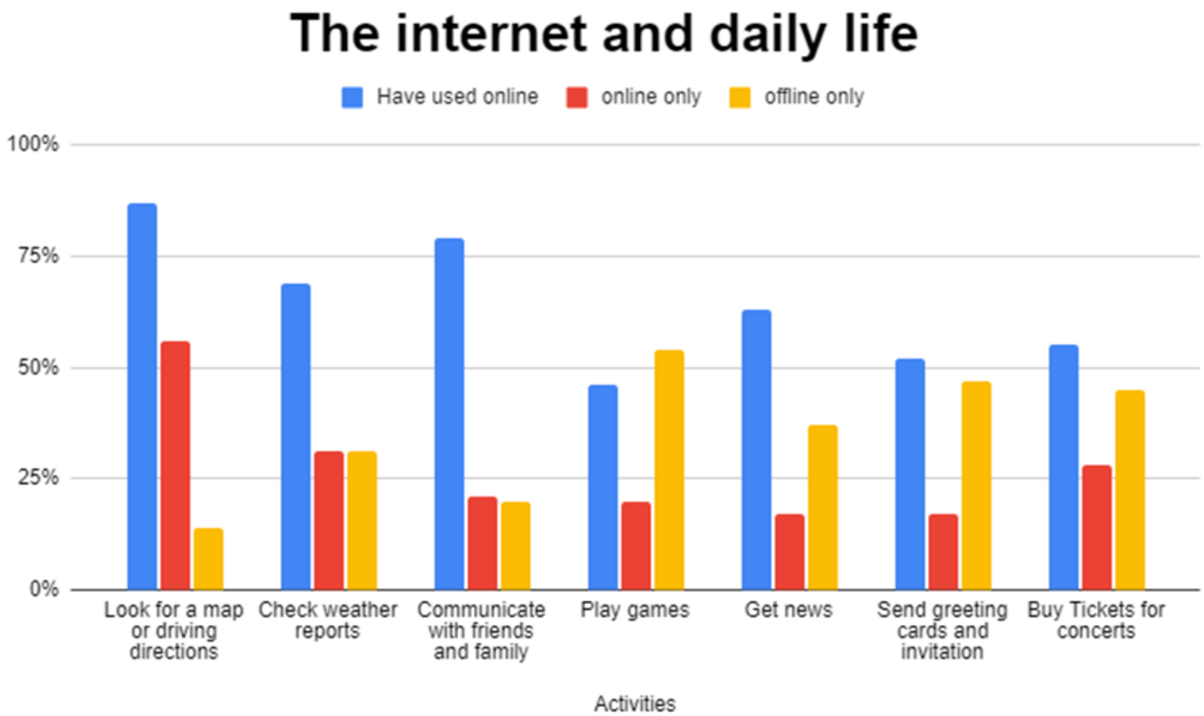


Figure 5. The internet and daily life. This figure presents statistics of how the internet has changed people's lives (Adapted by Hu (2021) from Pew Research Center, 2004)

Today, the internet "offers an immense wealth of possibilities for buying content, news, and leisure products, and all sorts of advantages arise from e-commerce, which has become a major distribution channel for goods and services" (Dentzel, 2014, p. 38). As a result, the internet contributes to the economy significantly. For example., the foundation of e-commerce has boosted businesses worldwide, and sales are expected to grow to 4 trillion dollars by 2020. This encourages the growth of small to medium-sized companies (Sharma, 2018).

Furthermore, many things people do traditionally are fundamentally altered by the internet. For example, signing a signature has been a longstanding tradition because it represents a person on essential documents and tells others that it is you and nobody else (Kobi, 2017). However, in recent years, cloud service has become popular, and e-signature has been created. DocuSign, an online provider for e-Signature that documents and manages electronic agreements, was founded in 2003 and has facilitated secure digital signatures (Kriz, 2018). This new digital signing platform has changed the way people deal with signing signatures. Instead of using a physical pen, people can tap on the screen and craft their signatures. This has influenced society as signing signatures has become much easier and more secure (Perucci, 2019).

WHY INTERNET IS REVOLUTIONARY

It is undeniable that the internet has drastic impacts on society and how it fundamentally changes our lives. However, few works have studied why the internet is so revolutionary. Specifically, it remains unclear why the technology of the internet can grow so fast. This is an important question because understanding the internet can help us understand the world better as the world is changing rapidly. Moreover, it can serve as guidance to us as we step into the next era of AI and metaverse.

This research will investigate the question and find possible explanations. The research hypothesizes that the internet becomes so revolutionary not because of its internal technical logic but because of its external interactions with the social world. Hence, the research will use the Social Construction of Technology (SCOT) framework proposed by Trevor Pinch and Wiebe Bijker (Bijker & Pinch, 1984) as the primary approach.

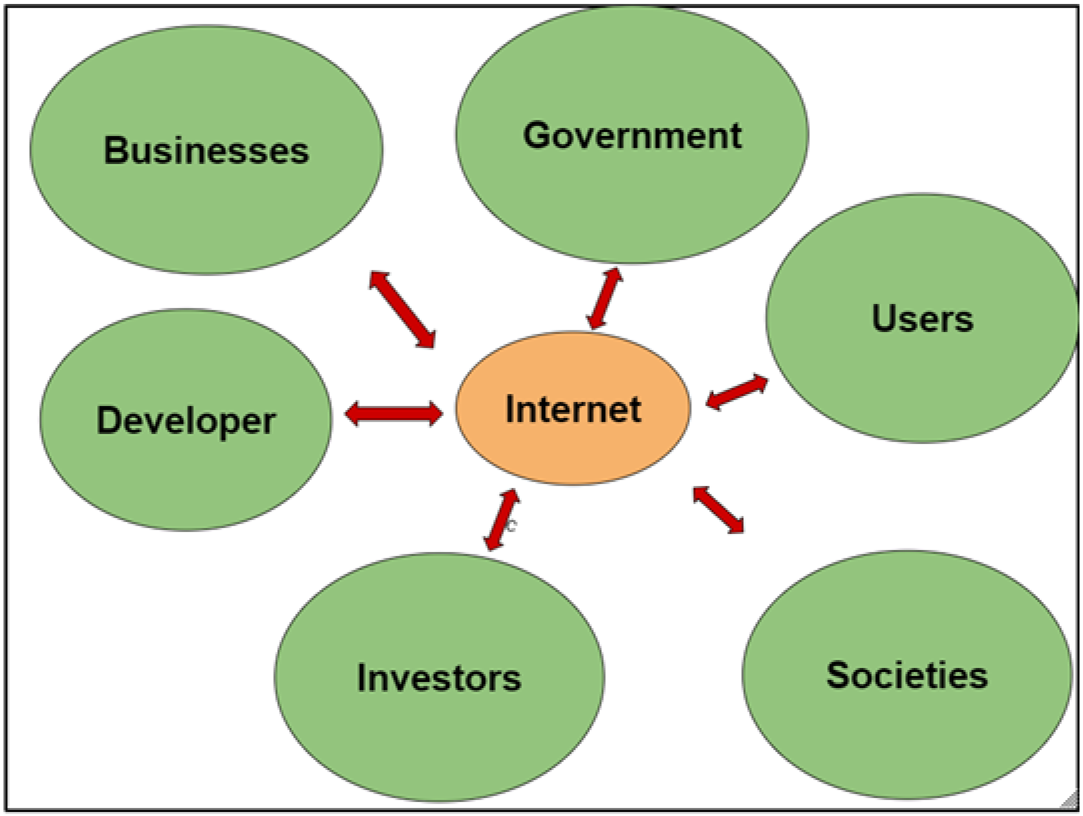


Figure 6. Revolutionary Internet SCOT model. The internet interacts with each social group and gets feedback to update itself. (Adapted by Hu (2021) from Carlson, 2009)

The Social Construction of Technology (SCOT) argues that technology does not determine human action but instead human action shapes technology (Bijker & Pinch, 1984). Therefore, in this system, presented in Figure 6, the internet is the center of the SCOT framework, while key relationships are business, government, developer, users, investors, and societies. Because the internet is a free and open space (The White House, 2014), each party can freely provide a perspective that will shape the characteristics of the internet. Subsequently, the internet evolves to reflect the interests and concerns of each party. For example, an individual can share his/her knowledge by writing a blog on the internet. This contributes to the evolution of the internet because the internet becomes more useful to some other people who seek this knowledge. The internet satisfies the needs of both parties: the party that wants to share and the party that wants to learn. Thus, by studying how these interactions between the internet and each party contribute to the evolution of the internet, it is possible to draw insights to explain why the internet is revolutionary.

This STS research project will be in the form of a scholarly article outlining why the internet is revolutionary. It will attempt to use the SCOT framework to seek to understand the reasons for the success of the internet from a social context. Most importantly, it aims to shed light on understanding today's world and the relationship between the internet and society better.

REFERENCES

Bijker, W. E., & Pinch, T. J. (1984). The social construction of facts and artifacts. *Social Studies of Science, 14, 399–441*. https://doi.org/10.1177/030631284014003004

Dentzel, Z. (2014, April 01). How the internet has changed everyday life. *BBVA OpenMind*. https://www.bbvaopenmind.com/en/articles/internet-changed-everyday-life/

Grossman, D. (2018, April 30). On this day 25 years ago, the web became public domain. *Popular Mechanics*. https://www.popularmechanics.com/culture/web/a20104417/www-public-domain/

Guerin, L. (2021, n.d.). First-day paperwork for new employees: understand what you're signing. *NOLO*. https://www.nolo.com/legal-encyclopedia/first-day-paperwork-new-employees-33052.html

Henry, M. (2019, June 24). Letter-writing 101: how to address an envelope. *Today*. https://www.today.com/home/how-address-envelope-t156576

Hu, Z. (2021). Automatic signature machine workflow. [Figure 2]. Prospectus (Unpublished undergraduate thesis). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Hu, Z. (2021). Gantt chart UVa automatic signature machine capstone. [Figure 1]. Prospectus (Unpublished undergraduate thesis). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Hu, Z. (2021). Revolutionary internet SCOT model (Adapted from Carlson, 2009). [Figure 6]. Class handout. (Unpublished). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Hu, Z. (2021). The internet and daily life (adapted from Pew Research Center, 2004). [Figure 5]. Prospectus. (Unpublished). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Hu, Z. (2021). What is waterfall methodology in software development? (adapted from Sundar, 2018). [Figure 3]. Prospectus. (Unpublished). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Kaysen, R. (2018, July 06). Buried in paperwork. *New York Times*. https://www.nytimes.com/2018/07/06/realestate/buried-in-paperwork.html

Kearns, P. (2019, September 19). Different periods of industrial revolution. [Figure 4]. PASCAL *International Observatory*. http://pascalobservatory.org/pascalnow/blogentry/news/being-human-era-fourth industrial-revolution-and-ai

Knight, S. (2018, August 30). Signing legal contracts: does a signature need to be in cursive? *Law Depot*. https://www.lawdepot.com/blog/signing-legal-contracts-does-a-signature-need-to-be-in cursive/

Knott, L. (2017, July 06). Repetitive strain injury. *Patient*. https://patient.info/bones-joints-muscles/tendinopathy-and-tenosynovitis-tendinosis/repetitive-strain-injury

Kobi, M. (2017, May 25). The importance of a powerful and beautiful signature. *KOACHMAN*. https://www.mrkoachman.com/importance-of-a-powerful-and-beautiful-signature/

Kriz, W. (2018, July 11). What is DocuSign and how does it work? *Merchant Maverick*. https://www.merchantmaverick.com/what-is-docusign-how-does-it-work/

Legalesign. (2016, February 19). The history of the signature. *Legalesign*. https://legalesign.com/blog/history-of-signatures/

Orem, T. (2021, October 04). How to fill out a new W-4 in 2021: guide and FAQs. *Nerdwallet*. https://www.nerdwallet.com/article/taxes/how-to-fill-out-form-w4-guide

Perucci, D. (2019, May 22). The importance of e-signatures: 8 reasons to make the switch. BambooHR. https://www.bamboohr.com/blog/importance-of-esignatures/

Petrow, S. (2018, April 11). Why signature matters. *The New York Times*. https://www.nytimes.com/2018/04/11/opinion/credit-card-signature.html

Schwab, K. (2016, January 14). The fourth industrial revolution: what it means, how to Respond. *World Economic Forum*. https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/

Sharma, V. (2018, April 25). Importance of ecommerce in our daily life. *Klient Solutech*. https://www.klientsolutech.com/importance-of-ecommerce-in-our-daily-life/

The White House, (2014, November 10). Net Neutrality. *The White House*. https://obamawhitehouse.archives.gov/net-neutrality

Texas Instrument, (2018). MSP-TS432PZ100. MSP432™ SimpleLink™ Microcontrollers *Hardware Tool*s. https://www.ti.com/lit/ug/slau571d/slau571d.pdf

Thomas, D. (2019, August 22). What is the digital age? *Ventiv Technology*. https://www.ventivtech.com/blog/what-is-the-digital-age