eTextile Interface for Generative Design Software

A Major Qualifying Project submitted to the faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements for the Degree of Bachelor of Science.

Submitted by:

Jenna Charron
Electrical and Computer Engineering

Kyle Mikolajczyk
Computer Science

Noelle Morgan
Electrical and Computer Engineering

Project Advisors:

Professor Shamsnaz Bhada
Electrical and Computer Engineering
Worcester Polytechnic Institute

Professor Gillian Smith
Computer Science/IMGD
Worcester Polytechnic Institute

Submitted on:
March 25th, 2022

Abstract

The goal of this project was to design a device that could assist novice to intermediate quilters in their creative design process by allowing users to visualize their designs digitally while maintaining the more familiar, tangible actions of creating on a design wall. We created a prototype based on the existing live web-based quilting software, Code Crafters, with the added element of a physical design mat that communicated using the latest WebUSB microcontroller communication standard. We used web-based software alongside the design mat to detect changes from the user and reflect them in software. We built this system using the Arduino Leonardo microcontroller, embroidered conductive thread, an LED matrix, and an innovative design of multiplexers and magnets to determine patch orientation. This design met the team's success criteria of Software Communication, User Interaction, and Design Functionality that was derived from system engineering concepts.

Acknowledgments

This project could not have been successful without the support and resources of several individuals in the Worcester Polytechnic Institute community. We would like to thank Professor Gillian Smith and Professor Shamsnaz Bhada, our advisors, for their support and guidance throughout the design and prototyping of this project. We would also like to thank Professor Charles Roberts for his guidance with communicating between the controller and the web interface. Lastly, we would like to thank Alison Darling for all her help with ordering and acquiring parts for our prototyping.

Authorship

Chapter Title	Author(s)	Editor(s)
1.1 Motivation	JC	All
1.2 Goal of Project	All	All
2.1 Brief History of Entertainment and Simulation Game Controllers	KM	All
2.2 Summary of eTextiles	NM	All
2.3 Current Technology	NM	All
2.4 Market Research	JC	All
3.1 Stakeholders and System Needs	JC	All
3.2 System Requirements	JC	All
3.3 High Level Design	NM	All
3.4 Network Topology/ Network Approach	KM	All
3.5 Technical Challenges	KM, JC	All
3.6 Project Timeline	JC	All
4.1 Component Trade-off Studies	All	All
5.1 Hardware Design	JC, NM	All
5.2 Software Design	KM	All
5.3 Hardware and Software Integration	KM	All
6.1 Research Methods	JC	All
7.1 Paper Prototype Testing Results and Feedback	JC	All
7.2 Prototype Testing Results and Feedback	JC	All
8.0 Conclusions and Future Work	All	All

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	**Description**
IoT	Internet of Things
EPA	Environmental Protection Agency
AI	Artificial Intelligence
NES	Nintendo Entertainment System
UI	User Interface
LED	Light-Emitting Diode
USB	Universal Serial Bus
TFT	Thin Film Transistor
OLED	Organic Light-Emitting Diode
LCD	Liquid Crystal Display
RGB	Red Green Blue
SRAM	Static Random-Access Memory
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
TLS	Transport Layer Security
CA	Certificate Authority
CSA	Certificate Signing Authority
CSR	Certificate Signing Request
JSON	JavaScript Object Notation
DNS	Domain Name System
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
OOP	Object Oriented Programming

1.0 Introduction

Integrating electronics into clothing has long since been a dream of the future, from Queen Elizabeth I's intricate and genuine gold and silver goldwork embroidery to Marty McFly's 2015 outfit in *Back to the Future*. With current technology, eTextiles (electronic textiles) have extraordinary potential within the medical field, art projects, fashion, and many more areas yet to be discovered. In this project, we explored the design and fabrication of a soft fabric controller to interface with a generative design software that simulates a quilting design wall. In this report, we will present the process and information that went into the design and fabrication of this prototype. We will begin in this section with the purpose of this project and our overall goals for this project. Then, we will present background information that we used to better understand our problem and form an appropriate solution. Following this, we will explain how we used systems engineering concepts to determine our system requirements, measures of success, and lay out a timeline for our project work. Moreover, we will further detail the technical components used in the final software and hardware design of our prototype. Then, we will present our testing protocols and the results of these tests. Finally, we will present our analysis of these results to make final conclusions and recommendations for future iterations of this project.

1.1 Motivation

Quilting has historically been a very physical process, in both the design phase as well as the creation phase. Quilters typically plan out their designs through design walls, which were typically made by attaching small pieces of fabric with pins to a larger piece of felt or fabric draped over a wall. The quilter moves the pieces around to test and create a quilt design before beginning work on the final quilt (The Spruce Crafts 2022). These “walls” allow quilters to mix and match designs before actually “quilting”, allowing them to visualize their patterns before committing to it. Unfortunately, design walls have a lot of drawbacks, with them requiring plenty of space, materials, and being difficult to move or share with others. A normal design wall is not environmentally friendly because of the amount of fabric that is required and wasted in the process. In addition, the lack of modern ubiquitous technology and shareability between peers that many younger generations have grown to appreciate and expect could pose an issue in the quilting world and potentially cause a decrease in interest in the craft. With this in mind, we were motivated to find a solution to these issues by making a virtual representation of design walls through the use of a controller and an interactive web tool.

1.2 Goal of Project

The goal for this project is to create a soft fabric controller to interface with a generative web design software that simulates a quilting design wall. While design walls have traditionally been used by experienced quilters and are proven to work, this project aims to improve and modernize this practice with the integration of eTextiles and generative software through the creation of a controller.

Figure 1.1: Example Quilt (Wikimedia Commons 2022)

Traditional design walls typically take up a lot of space and are not easily portable due to their size and the amount of material involved with using them. While there are some ways to make portable design walls, such as by using poster board or vinyl tablecloths, integrating the design process with software will allow less of a dependency on space and allow more portability (The Spruce Crafts 2022). Integrating the design wall process with software can also reduce fabric waste since the software can help a quilter visualize many designs without the need for new fabric. In 2018, the Environmental Protection Agency (EPA) estimated that 11.3 million tons of fabric waste ended up in landfills whereas only 2.5 million tons of fabric waste was recycled (US EPA 2017). Fabric waste is becoming more prominent amongst environmentally conscious consumers, therefore textile markets, including quilting, will need to adapt to more environmentally responsible processes and products.

Our project intends to make designing quilts less resource dependent and more cost effective and environmentally friendly, which in the long term goal being to make quilting to be more accessible to other non-typical markets such as those with lower income for purchasing the necessary materials. In 2020, Premier Needle Arts funded a survey of quilters and a demographic they collected in this survey was the employment of quilters. . From this survey, 64.3% of quilters are retired individuals who have limited funds to devote to quilting. Our product could financially help this demographic (Craft Industry Alliance 2020).

The generative design software that this project is paired with, CodeCrafters, is a tool designed by researchers at Worcester Polytechnic Institute and Georgia Institute of Technology to allow quilt designers to add random generation into their creative process, increasing the variety and possible designs they can produce. This project provides a tactile interface that is relatively familiar to experienced quilters for this generative design software and gives users the freedom to pick and choose parts of a random generation that they want to build off of. This modernized, technology centered approach to a traditional fabric design process allows for more freedom to iterate and visualize through many designs and assists in inspiring new ideas through the generative nature of the software and device.

2.0 Background

This project integrates technology from video game controllers and eTextiles to create a device that can serve the quilting design community. The following chapter serves to provide the reader with the necessary background knowledge needed for the development of this project and for further understanding of chapters to follow. These sections provide background on the history of entertainment and simulation game controllers, a summary of eTextiles applications, current eTextiles technology in development and on the market, and research on market trends relevant to this project.

2.1 Brief History of Entertainment and Simulation Game Controllers

Controllers have always been the way that people interact with the virtual world. Controllers translate human interaction with a physical device and convert it through sensors into analog and digital signals that the computer can understand. Although the first controllers were designed for playing video games for entertainment, over the decades more and more controllers have been designed for purposes that far outreach what the first controllers were designed for. As computers became more advanced, designers began to realize that controllers can be used for simulating tools in the real world. While modern game controllers are still designed for people to play fun video games with, controllers have been designed to simulate cars, airplanes, and other real world objects and experiences. Of all the controllers designed over the years, there are a select few that have provided excellent inspiration when looking into designing an eTextile quilt designing controller.

Often, the Atari 2600's joystick is considered to be the first widely-used entertainment controller that worked for many games. This joystick consisted of a paddle that could move in four directions, and a button. The simplicity and versatility of this joystick allowed for nearly 500 games to be developed for the system since the console's release in 1977 (Farquhar 2020). This controller was designed for entertainment, as it did not emulate or look anything like a real object. The 2600's joystick was designed to simply respond to user feedback, and it did that extremely well. When designing a controller, the most widespread and useful controllers have been simple in the design and User Interface (UI).

Figure 2.1: Atari 2600 Controller (Wikimedia Commons 2022)

When the Guitar controller for the first Guitar Hero video game was released, it sold one billion dollars worth of product in the first week alone. The controller was even designed to look very similar to an actual electric guitar at ¾ scale. However, the controls were simplified to make the user experience and game more enjoyable, even to people who didn't know how to play a guitar. Primarily, the strings were replaced by 5 colored buttons with a button that the user could "strum" similarly to an actual guitar. Thousands of people were then able to experience the thrill of playing a guitar, to music, using a simple game controller without needing prior experience. (Hardy 2014). This controller was designed to simulate the look and feel of a real guitar, but removed the difficulty of playing by replacing the strings with buttons, similar to those found on conventional game controllers. It was designed for entertainment, allowing users to easily adjust from a real-life experience of playing a guitar to playing a virtual guitar with ease.

Figure 2.2: Guitar Hero Game Controller (Wikimedia Commons 2022)

The first widespread entertainment game controller that was designed for a very specific purpose was the Nintendo Entertainment System's (NES) Zapper in 1985. This controller, unlike previous widespread competitors, was designed specifically to be used as a gun in a video game. The controller consisted of only one button, the trigger, and the user had to aim the Zapper at the TV screen. The controller was also shaped to resemble a gun, and usage was similar as well. This allowed for both a new way to play video games, and for the user to feel more involved in the game. Just like in real life, users had to practice aim if they wanted to do well, instead of moving a paddle (Barker 2007). Although this controller was designed for entertainment, it actually simulated a real gun very well while removing the learning curve. There was no user feedback (such as vibrations, resistance) when the user used it, but other mechanics would translate well to a paintball gun. The Zapper proved that designing a controller for a specific purpose not only sold well but enhanced the user's experience greatly.

Figure 2.3: NES Zapper (Wikimedia Commons 2022)

Before 1997, driving a car in entertainment video games was controlled using unresponsive game controllers, as nearly all games were designed to use. That all changed when Microsoft released the Sidewinder Force Feedback Wheel. This controller was revolutionary in that not only did it allow users to use their feet to control the game (the pedals) but the steering wheel, which resembled and felt like a car's steering wheel, had motors to provide feedback and resistance when the user used the wheel (Cornish 2021). This controller was designed to be a serious tool in racing video games. People wanted the full physical feel of turning a steering wheel and to see the interaction on the screen mimic what they would see when driving a car. By practicing on the virtual car, the experiences were designed to translate to the real world. Sending feedback to the user was critical for the controller to allow the user to react to feedback in real time.

If consumers want an experience with controllers to be as realistic as possible, there are manufacturers that create controllers near-identical to the real thing. An example of this is the HOTAS Warthog joystick and throttle. This controller looks and feels as if it came directly out of a fighter jet. It comes equipped with 55 programmable buttons, a full throttle and joystick (Thrustmaster 2021). It is designed for people who want to simulate flying a plane, without the exuberant costs of owning a real one or getting a license. These controllers become tools that can help users practice or do maneuvers that would otherwise be inaccessible to them.

There have also been forms of controllers that have never made it to market for various reasons. One controller that was demoed but not released was the Nintendo Knitting Machine. The premise was that the user could use the controller in conjunction with the Nintendo Entertainment System to knit (Majaski 1987). Interestingly, in more recent years, the same company began to produce controller kits that could bring virtual games to real life. Named “Nintendo Labo”, these kits are sold as accessories to Nintendo’s current game console: “The Nintendo Switch”. These kits are simple cardboard cutouts and other craft supplies sold with the software necessary to use it in conjunction with the console’s existing controllers, the “Nintendo Joycons”. This product allows for the controllers to adopt entirely new functions ranging from piano keys to a race car steering wheel (Fox 2018). Although the Nintendo Knitting Machine and Nintendo Labo systems had different goals, they both aimed for software tools to have a real-world presence, and proved that virtual tools can assist with and impact learning and designs in the real world.

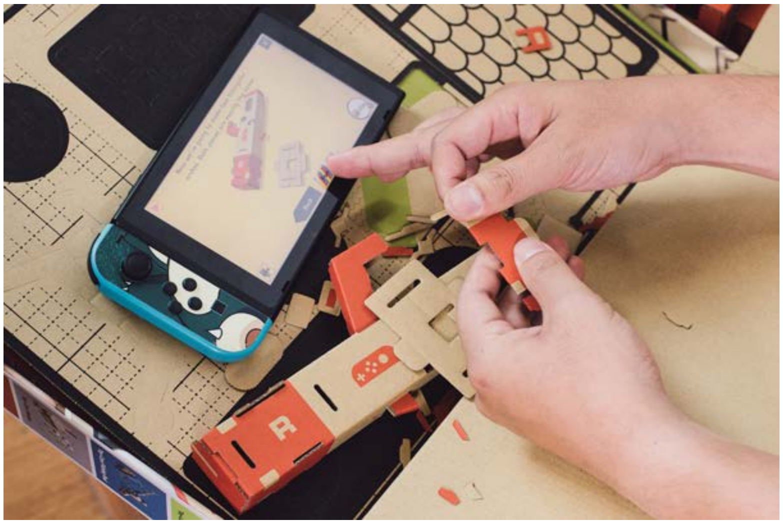

Figure 2.4: Nintendo Labo Kit (Wikimedia Commons 2022)

These various controllers will serve as the backbone of the design for a great controller. Using aspects of each, such as the simplicity of the Atari 2600's controller, the Guitar Hero's and Nintendo Zapper's resemblance to real-life objects, and the Microsoft Sidewinder's tactile feedback has the potential to result in a controller that users can find helpful, easy to use, and allows for the digitalization of a previously only physical tool.

2.2 Summary of eTextiles

A more modern and recognizable application of eTextiles was the invention of core rope memory. This technology was first used in the early NASA Mars space probes and then in the Apollo Guidance Computer (Brock 2017). Core rope memory was a method of storing data (i.e. a program) that represented individual bits as threads of wire and magnetic beads. These were woven together in a way that formed a readable computer program. Since then, eTextiles have evolved to use more fabric-based and wearable mediums. The following sections outline examples of these wearable, fabric-based applications of eTextiles.

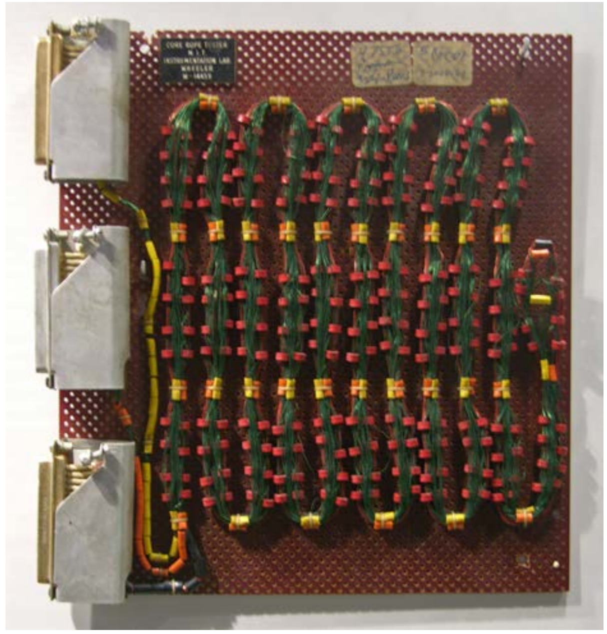

Figure 2.5: Core Rope Memory Test Sample from Apollo Program (Wikimedia Commons 2022)

Bleeding Edge Research

Much of the eTextile technology development taking place today can be found taking place at many universities and facilities around the world. In addition to these more established technologies, experimental ones are in development as well and are even available to the average person. One example of relatively more mature eTextile technology being developed in a research setting would be that done at Ohio State University.

At Ohio State University, Researchers have successfully embroidered circuits into fabric with a precision of 0.1mm. They've also developed methods to embed RFID chips into their embroidery and investigated ways to add color to this new form of embroidery to make the technology more useful for logos and colorful designs. They hope to use this technology to gather, store, or transmit digital information through functional textiles. Other applications include antennae for other devices, vital measurements, and even sensing brain activity (Kiourty et al 2015).

Another example of eTextiles in research includes the use of knitting machines to create a fabric based keyboard . In the research documented in the project, researchers were able to knit circuits directly into the textile substrate, creating a flexible soft keyboard (Wicaksono, Paradaiso*, et al,* 2020). Industrial knitting machines were programmed specifically for this project and both conductive and non-conductive yarns were used to sense when a user touched a key. This is just another example of the versatility within eTextiles: the electronic components are not limited to just being sewn or embroidered.

UV Fiber Optic Pajamas for Jaundice Treatment (By Empa)

Another novel example of the application of eTextile technology is EMPA's development of phototherapeutic pajamas to treat infants with jaundice. Their particular innovation involves fiber optic filaments woven in a precise pattern so that items "made with it would only emit blue light inward, directly onto the child's skin" with consistency and over a fairly large surface area (Waltz 2017). This product is a significant demonstration of how valuable eTextiles are within the medical field because it prevents newborns from having to be alone in light-filled incubators, which is the standard approach to treating jaundice, and allows parents to comfortably hold their children during particularly trying times.

Figure 2.6: An Illuminated Blanket Designed To Treat Jaundice (Wikimedia Commons 2008)

Flora Pads

Outside of these specialized fields, the average person can also experiment with eTextiles. With development boards such as the Lilypad Arduino shown below and the FLORA wearable platform, hobbyists and "Makers" can make their own creations with relatively easy to acquire materials.

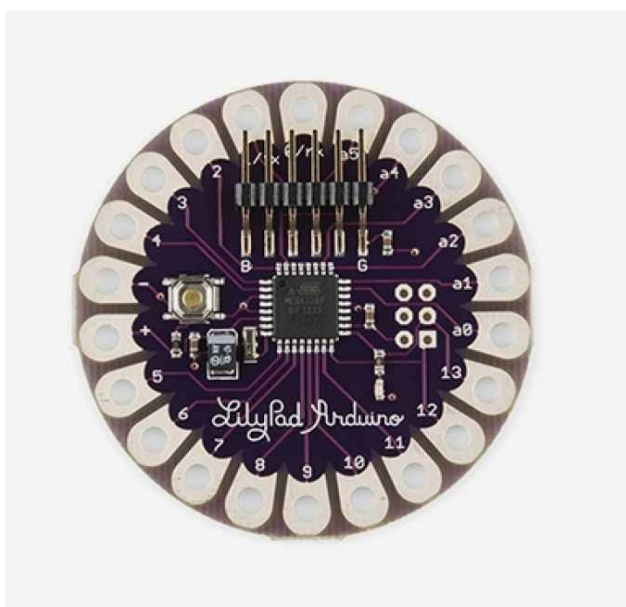

Figure 2.7: Lilypad Arduino (Arduino 2022)

Electrical connections are made using conductive threads and either traditional through-hole components with curled leads, or specialized circuit boards with holes big enough to sew through. The PCB has large holes that make them easy to sew through by hand, and is small enough to attach to a garment or relatively small project (Arduino 2022). The use of the Arduino platform makes controlling the electronics tremendously accessible due to the huge community around the platform.

There is a considerably large overlap between art and eTextiles, as seen in the broad range of applications for this field of interest. One area where eTextiles might be particularly useful would be among quilters, who deal in textiles on a regular basis. Fabric is a familiar material to this community making them an excellent target market for an eTextile based device.

2.3 Current Technology

The current eTextile technology available to the average consumer is highly focused around the hobbyist market. Websites such as https://www.adafruit.com/ and https://www.sparkfun.com/ are great sources for components and support for hobbyist projects. However, the pool of materials and knowledge available to the public at the time of writing this is rather small because current eTextile technology is still in its infancy.

The Flora and Lillypad Arduino, as mentioned earlier, are sewable microcontrollers meant to be used alongside conductive thread. While they are highly documented and make prototyping incredibly easy, their wide and flat shape are not ideal for an item designed to be rolled up, and could even snap if the end user isn't careful enough. In addition, these development boards do not have the required memory capacity, as will be mentioned below.

Despite this, the large sewable holes used in these products can provide inspiration for future circuit board designs used in more refined versions of our system (Arduino 2022).

Another example of current eTextile technology would be Loomia's wide range of eTextile products. These flat, flexible sensors are already designed to be used in eTextile applications and are being used in projects run by companies like Hyundai, Analog Devices, and more. For use in our design, however, they are not cost effective, nor are they available in the specific sizes required for this project. Our controller required a very specific form factor, which Loomia products already have predefined, so we studied the products for inspiration regarding available sensors and then designed custom sensors for our controller.

2.4 Market Research

This project aims to provide a solution to aid quilters in the design process. To research a solution, we looked into market trends of the quilting and eTextile industries to supplement our research on controllers and eTextile technology. Recent market trends have indicated an increase in eTextiles on the market. A majority of these smart textiles on the market use optical fibers, conductive metals, and polymers to interact with the environment around them. Despite the COVID-19 pandemic impacting the supply chain and decreased textile consumption, the smart textiles market is projected to recover and continue its previous growth in the coming years. However, the smart textile market will need to address increasing concerns about the environmental impacts of producing along with the life cycle of smart textiles to meet the market demand for more sustainable textile options (Grand View Research, Inc 2021).

From 2021 to 2026, the smart textiles market is projected to grow from 2.3 billion USD to over 6.6 billion USD. This growth is primarily attributed to the explosion of Artificial

Intelligence (AI) and Internet of Things (IoT) technology, along with the growth in wearable electronics in recent years. The growth of the smart textiles market is predicted to be driven by developments in applying advanced technologies in smart textiles, increased availability of smaller electronic components and wearable electronic components (Market 2021).

Figure 2.8: Example of eTextiles, LEDs Built into a Dress (Wikimedia Commons 2022)

In addition to that, a major consumer demand for eTextiles came from demand for fitness apparel or accessories with smart sensing and monitoring capabilities during the COVID-19 Pandemic because of the increase in the population working from home. Textiles that are integrated with electronics that are designed to sense and react to the environment, also called active textiles, made up the largest market share of the smart textiles segment in the year 2020. The greater interest in smart textiles due to the COVID-19 Pandemic is expected to aid in boosting the eTextile market in the present and future. These smart textiles have applications in many fields including medical, sports, and military. One restraint to the growth of the market, however, is the lack of regulations and standards for smart fabric manufacturing as well as the potential for environmental and health hazards associated with the production of eTextiles (Market 2021).

Given that this project intersects eTextiles with quilting, the interests and strength of the quilting market was an important factor to consider in the research and design of this product. The 2020 survey, funded by Premier Needle Arts that was sent to many quilters, provided insight on the demographics of our devices' potential user base. The results indicate that the target market primarily quilts to relieve stress, be creative, and connect with family and friends (Craft Industry Alliance 2020). Our project goals relate directly to improving the creative process and connections with the community for quilters.

Projections of the quilting industry show that the market will continue to thrive in the coming years. Quilting has grown in popularity from a 10% production capacity in 2017 to about 15% production capacity in 2020. Additionally, the market has begun to shift towards digital and online platforms. Many quilters have begun to apply digital design applications to their quilt designs and shop online rather than in brick-and-mortar stores (Market Research Blog 2021). Our project satisfied the trends towards a more digital craft. The current strength and future growth of the eTextile and quilting markets indicate that our project has a place in the industry.

3.0 Concept of Operations

This chapter helps guide the design process of evaluating the stakeholders for this project and identifying and prioritizing their needs to develop a set of system requirements and a system design. This chapter explains these stakeholder needs, system requirements, a high level system design, the network topology and the timeline for completion of this project.

3.1 Stakeholders and System Needs

In developing a device to achieve our project's purpose, it was important to identify and prioritize the individuals and entities that have an interest in our proposed solution. These identified stakeholders include users, the project's team, its advisors, maintainers, producers, and sellers as identified in Table 3.1.

Table 3.1: System Stakeholders

ID	**Title**	**Description**	**Influence**	**Needs**
SH.01	Users	Individuals use this device for quilt design.	Medium	Strong
SH.02	Project Team	The student team who have designed and built the prototype for this device as a capstone project to their education.	High	Weak
SH.03	Project Advisors	The advisors of the student project who oversee the progress and adequate completion of the project in line with degree requirements.	Medium	Weak
SH.04	Maintainers	Individuals who will perform any necessary maintenance on the system.	Low	Strong
SH.05	Producers	Individuals/organizations who would produce this device to introduce into the market.	Low	Medium
SH.06	Sellers	Organizations who would sell this device to users.	Low	Strong

The motivation behind this project was to create a virtual quilt design application by using a controller to communicate with a web interface to enhance the design process for individuals looking for a digital method to design quilts. These individuals are represented by the users and are one of the highest priority stakeholders. Users have a medium influence level on the design as they are the “end-user” of this device and have strong needs that are incorporated into the design of the system.

Next, the project team and advisors are stakeholders responsible for the design and development of the system. These stakeholders have a high influence on the design but low priority needs for the system. Both the project team and advisors are interested in satisfying the needs of the users and creating a system that is functional and fosters the application of skills and knowledge from their respective degree programs.

The external stakeholders are the maintainers, producers, and sellers of the controller. These stakeholders become relevant if this system goes to market and will need to be maintained, go into production, and be marketable, should the system be profitable. These stakeholders have a low influence on the design of this system, but have important needs that must be taken into account by the developers. By considering the needs of the identified stakeholders, we derived several system specific needs to satisfy the implicitly and explicitly stated requirements of the stakeholders. These system needs include marketability, user interaction, software communication, community and design functionality as presented in Table 3.2.

Table 3.2: System Needs

ID	Title	Description	Priority	Complexity	Traceability
N.01	Marketability	The system should be appealing to the appropriate market and fulfill the needs of the market.	Low	Moderate	SH.01, SH.05, SH.06
N.02	User Interaction	The system should be easy for the user to learn how to use and have a familiar user interface for quilters.	High	Moderate	SH.01,SH.02, SH.04
N.03	Software Communication	The system should be able to successfully communicate between the web interface and the controller.	High	High	SH.02, SH.03, SH.04
N.04	Community	The system should have capabilities that allow users to share their designs and see other quilters' designs.	Low	Low	SH.01
N.05	Design Functionality	This system should have the necessary functionality to enable the quilting design process	High	High	SH.01, SH.02, SH.03, SH.04

As explained in Section 2.5, there is an increasing market for eTextiles, with this device providing a solution to incorporate those eTextiles into a more convenient and effective method of quilt design. Therefore, one of our system needs is that our system is marketable. The system should be appealing to quilt designers looking for a product to help them digitally design quilting patterns. To the project team, however, this is a low priority system need because functionality was deemed more important. The low priority designation means that other system needs will be focused on before this one.

Our project's motivation and goals outlined a need for this system to have a familiar interface for quilters to work with. With eTextiles, we will be able to provide this familiar fabric interface that allows users to easily learn and work with the system. Familiarity will result in users being able to focus on their design more than the functionality of the system. As such, user interaction is a high priority system need that is moderately complex. The need traces back to improving the user's experience and those responsible for the systems' maintenance having an easier interaction with the system for any needed updates. Furthermore, this is also an interest of the project team to learn more about integrating eTextiles into a quilting application.

Another high priority for the system is software communication. Since the system consists of a complex web interface and a complex physical controller that needs to interact and send data between each other, it is vastly important that the software and hardware be able to send and receive data so that the system can function in the intended way. This is a complex aspect of the system's functionality and therefore a decidedly high priority in the device design and creation.

Moreover, an aspect that CodeCrafters already provides functionality for is the ability for quilt designers to share and browse each others' designs. This feature is a helpful element that allows the community to connect, share and learn together. Thus, shareability is a system need that will enhance the system's benefit to the user. Since this functionality already exists in the software, the system need is not complex. This component is of a lower priority due to the fact that it is not necessary for the system to function and serve its purpose, but would rather have it as an added element that enhances the user experience.

Finally, another important system need is for the system to function in a way that allows quilt designers to effectively and easily create designs. There are many requirements that go into allowing the system to function in this way, thus there is high complexity in this system need. However, it is integral to the success and effectiveness of this system that this need is met.

3.2 System Requirements

With an understanding of the stakeholder needs and priorities and how those translate into system needs, we are able to generate several system requirements that should be met for this system to satisfy the needs of these stakeholders. Each of these requirements are prioritized by the priorities of the stakeholders and can be traced to a specific need of the system. The system requirements are shown below in Table 3.3.

Table 3.3: System Requirements

ID	Title	Description	Priority	Verification	Traceability
F.01	Idea Generation	The system shall randomly generate design elements and allow the user to keep certain random elements.	High	Demonstrate	N.05
F.02	Copy/Paste	The system shall have functionality for the user to copy a patch or block and paste it to a new patch or block.	Medium	Demonstrate	N.05
F.03	Color Pallette	The system shall allow the user to visualize their color palette and make modifications to it.	High	Demonstrate	N.05
F.04	Visualization	The system shall have capabilities for the user to see their design on the web interface and roughly see aspects of their design on the controller.	High	Demonstrate	N.05
F.05	Patch Mobility	The system shall have patches that are easy to move and can be rotated to simulate a physical design wall process.	High	Demonstrate	N.05
F.06	Shareability	The system shall allow users to share their own designs with the community and see/use others' designs	Low	Demonstrate	N.04
F.07	Software Communication	The system shall communicate between the software and the controller and send and receive data	High	Demonstrate	N.03

		from the controller.			
F.08	Quick Learning Cycle	The system shall provide adequate instruction and be easy to use so that the user can quickly learn how to use the system.	Moderate	Test, User Feedback	N.02
F.09	Ease of Use	The system shall be easy for the user to interact with and provide feedback such as LED indications, instructions and error signals.	Moderate	Demonstrate, User Feedback	N.02
OP.01	Power Efficient	The system shall operate without a battery replacement for several active use hours.	Low	Model, Test	N.01
OP.02	Life Cycle	The system shall have a life cycle of at least a year.	Low	Model	N.01
P.01	Familiar Quilting Design Interface	The system shall use a fabric controller and allow the user to use familiar design wall movements to interact with the system.	Moderate	Inspect, User Feedback	N.02
P.02	Attractive Design	The system shall have a clean, attractive design.	Low	Inspect	N.01
P.03	Table-top Size	The system shall be small enough to conveniently fit on a table with a laptop.	Low	Inspect	N.01
P.04	Portability	The system shall be portable.	Low	Inspect	N.01

System requirements F.01 through F.09 are the functionality requirements of the system. Per the stakeholder priorities and system needs, these requirements include randomized idea generation, a copy and paste function, color palettes, visualization capabilities, moveable and rotatable patches, community shareability, communication between the software and controller, an easy learning curve, and its ease of use. These functionality requirements are prioritized based on the priorities outlined in the previous section on Stakeholders and System Needs as summarized in Table 3.3. Moreover, we will measure the success of most of these elements by demonstrating that the prototype has the specified capability. The learning cycle will then be measured through testing the time it takes for participants to learn how to use the prototype and by user feedback from participants testing the prototype. Lastly, the ease of use of the system will be measured by demonstrating the device's ability to give visual feedback to the user and by user feedback after testing the prototype.

Next, the requirements OP.01 and OP.02 are operational requirements of the system. These requirements focus on the product life cycle and battery life of the device, and these requirements are a low priority due to the early development stage of this project and the timeline for this iteration of the project. Thus, the requirements will be measured by modeling the battery life and product life cycle from data sheets.

Finally, P.01 through P.04 are physical requirements of the system. These requirements focus on the physical interface, design, size and portability of the device. Based on the stakeholder needs and priorities, the device should have a layout that is a familiar setup to quilters and is modeled after how one would interact with a design wall. Additionally, this device will make use of eTextiles to produce a more familiar medium to quilters. This requirement is of moderate importance and will be measured by inspection and user feedback. Some lower priority requirements are that the system have an attractive design, can easily fit on a table with a laptop and can be portable. Success of these requirements will be measured by inspection that the prototype meets the requirement.

3.3 High Level Design

In order to fulfill the system requirements we had set, especially system requirements F.01 through F.09, the main subsystems of this device can be categorized into four sections: user **input and output, control and logic, browser communication**, and the **webpage** that this device interacts with itself.

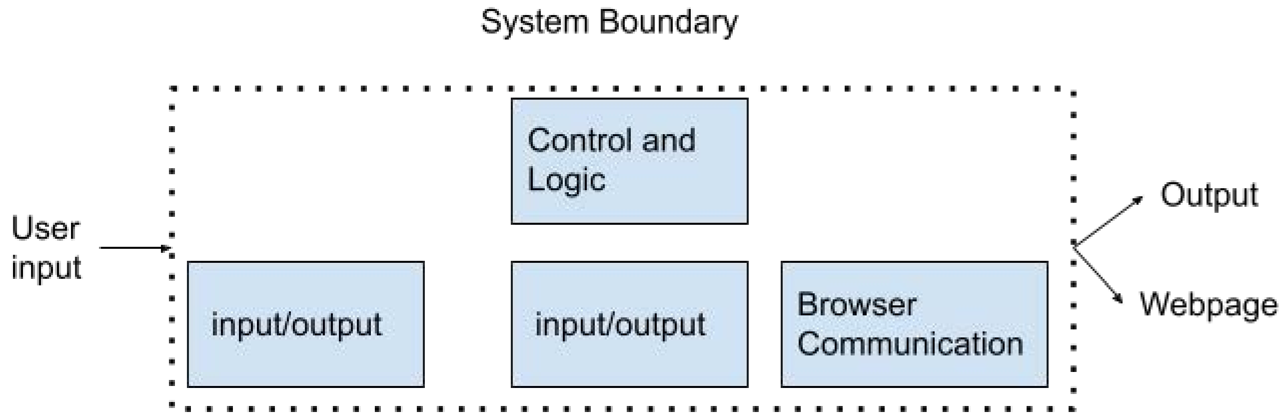

Figure 3.1: High Level Block Diagram, Broken Into Subsystems

3.3.1 Input and Output

The input and output subsystem consists of functions that accept and display inputs and outputs. These are imperative to system requirements F.05 and F.09 as well as the normal function of the device. Simpler inputs required to make the device functional include capacitive touch buttons and sliders that select and edit patch shapes and colors. More complicated information, such as the location and orientation of patches placed by the user is collected through a magnetic grid system. This achieves requirement F.05, allowing users to easily move and rotate patches to simulate a physical design wall process. Responses such as buzzers, haptic feedback motors, and the reactive LED display would satisfy requirement F.09. These functions indicate to the user that their responses have been received and recognized, with the LED display taking priority of all response devices.

3.3.2 Control and Logic

The control and logic subsystem ensures that the device interprets collected data and displays specific outputs in a consistent, predictable fashion. It also ensures that requirements

F.01 and F.04 are achieved as the idea generation and visualization are managed within this section. The majority of the logic being used is based on a simple reactive model. When the generate button is pressed, F.01 is fulfilled via the device's random patch generation ability. This, alongside any additional work the user has done, is interpreted and sent to be displayed, completing requirement F.04.

3.3.3 Browser Communication

Bi-way communication between the web tool and the controller will be possible through the WebUSB design that allows for a Universal Serial Bus (USB) device to be connected to a computer and communicate with the web browser. More specifically, the WebUSB can communicate over WebSockets, allowing for any data to be shared between the computer and the controller, in both sending and receiving on both sides (WebUSB Arduino 2022). They will pass JSON (JavaScript Object Notation) formatted strings, which can be quickly decoded to understand and execute the passed or given action or update. JSON is the best choice for communication, as JavaScript natively supports this form of data communication, and the majority of applications that communicate with JavaScript use this data type. This design will allow for the controler to be "plug-and-play", as in that there will not be a need for the user to download any software to use the controller. The user will be able to plug the controller into the computer, visit the site, and pair the controller with the click of a button, completing requirement F.07.

3.3.4 Webpage

The original CodeCrafters web tool needed to be modified to achieve F.02, F.03, F.06. The original CodeCrafters design's main focus was making the tool fully functional to the end user, with the user only being able to use a mouse and keyboard. The goal of making a copy/paste feature would help with the user's access and speed when using the physical controller, thus adding a button to the controller and displaying this on the webpage would satisfy F.02. One of the most important goals was F.03, which was adding a way for the user to see the color palette on the controller and the webpage. The webpage already has this feature on its screen, meaning the webpage would have to send this data directly to the controller, satisfying F.03. Lastly, the low priority goal of F.06 would allow the user to view the designs of others in real time, with use of interaction from the controller. The UI currently has the functionality to view the work of others, but because the UI is crowded with buttons and fields for functionality, it makes it difficult to use. The UI will need to be trimmed down and cleaned up to provide less functionality on screen, as this functionality will be translated over to the controller. The re-formatted UI would then satisfy F.06

3.4 Network Topology/ Network Approach

The tool CodeCrafters was written in a combination of HTML, CSS, and JavaScript languages as these are the core languages that all websites and web tools currently need to use to work properly on web browsers. More specifically, all the back-end of the tool is written with Node.js, which is a back-end JavaScript runtime that allows for powerful applications to be written with a back-end in JavaScript (Node.js 2021). This web-server will be run in the cloud, allowing for any user on the internet to use and interact with the software. Furthermore, the persistent data for live users and storage of works created by users will be stored using Google's Firebase, which is a realtime database in the cloud allowing for instantaneous updates and saving/loading of data (Google Firebase 2019).

For this controller, interactions will have to occur between the controller and the web browser. The web browser can locally send or receive data through Web Sockets, which is a way that a web browser can communicate both locally and remotely through sending data packets. Web Sockets allow for persistent communication without the need to stop communication at any time, and controllers require low latency and instant feedback which Web Sockets excel at (Websockets 2021).

3.5 User Stories

In order to get a better idea of how users will interact with our system and inform the importance of certain elements of our design, we created a few user stories to think through how different users would interact with our system. Below we examine five example user cases.

User Case 1

Background: Jane has been doing English paper piece quilting for a few years and feels fairly confident in their quilting knowledge but is still looking to learn more. She likes to design her own patches and put them together into a quilt. She has been doing her design on design walls and has not yet used design software. She is familiar with common computer applications such as internet browsers, spreadsheets and text editors but does not have any coding experience. She is looking for an easier way to see many different design ideas without needing the fabric for all of them so she is looking for a design application to help her with this.

Goals:

- Visualize ideas before moving to fabric.
- Has ideas and wants to see if they look good.

Potential Problems to Look-out For:

- Limitations of software on creativity.
- Changing habits to fit software.
- Design walls are not portable.

Behavior:

Enjoys interacting with fabric and the open creativity of quilting. More comfortable with physical interaction than computer interaction.

Approach:

- Has ideas for patches and wants to see how they work together before starting.
- Has quilting experience but minimal tech experience.
- Start with picking the color palette they have in mind.
- Designs patches using design space and copy / pasting (would benefit from clear instructions so the software doesn't hinder their quilting experience).
- Moves patches around and edits them on the board - similar to an actual design wall (would possibly use generate with a set of designed patches, but probably not random generate).

User Case 2

Background: Gertrude is a master quilter. She has sewn countless quilts for her friends and family, and won many awards for her awe-inspiring designs. She wants to use this tool to iterate potential designs.

Goals:

- Iterate quilt designs.
- Quickly record quilt ideas she may think up and want to save for later.

Potential problems to look out for:

- Would probably want more flexible design options.

Behavior:

Gertrude is a speed user. She already knows how to quilt. May or may not use the pattern generation tools, but would appreciate a lot of pieces to use. She would probably take it everywhere with her so that she can generate patterns on the fly.

Approach:

She is an expert quilter and would probably use this like a sketchbook for quilt designs.

User Case 3

Background: Jeff has recently gotten into Pieced Quilts quilting with his family. He has only been at it a few months, but has completed several small quilts for his friends in his computer science classes. They like to use them as mousepads, and he is realizing that there is money to be made here.

Quote: “My friends all love the one-of-a-kind quilts to use while programming or gaming”

Goals:

- To use the program and controller to generate quilt designs to help practice the art of quilting.
- To create quilt designs on the fly for his business.

Potential Problems to Look-out For:

- There is a lot of time in designing a quilt before putting it together.

Motivations:

- Saw quilts over the years that family had made him, and has always been curious on how to make them.
- Jeff also wants to make quilts to sell on his Etsy store and his own storefront, “quiltsRus.com”, to help pay off student loans and get his feet wet in running a profitable business.

Behavior:

Jeff is very good at creative solutions related to programming, primarily when designing backend servers for his popular website “quiltsRus.com”. Creativity for the UI of the website is the area he always struggles with, as he generally has to use templates if he wants people to enjoy the look and feel of the website.

Approach:

He gets requests for new quilts, specifically small-sized quilts for use as a mouse pad. The requesters give some input for what they want, such as colors they might like along with hobbies and other uniqueness to add to the quilt.Taking these ideas, he then turns to a tool and software to start designing patterns for use. He can also incorporate past patterns by other users or himself. Using the pad, he can quickly choose what he likes and doesn’t like from the generate function, until he has the design he likes. He can then transfer the design into the physical quilt.

<u>User Case 4</u>

Background: Glen is a tailor. They make formalwear professionally, and are mildly familiar with things like knitting and crochet. However, they haven’t ventured into the world of quilting. They’re looking to pick up a new hobby, and hope it may inspire new clothing designs.

Goals:

- To learn the basics of how to design a quilt.
- Rapidly iterate quilt designs before actually making the quilt.

Potential Problems to Look-out For:

- Square based quilts are the only option in the software, but may not always be the best shape for their pieces. Weirdly shaped quilts (octagons, swirls, other nesting shapes) might be appealing to them.
- Glen would also be interested in a free motion pattern generator to apply to their quilts as well, to allow more design options.

Behavior:

- Glen would likely attempt to learn as much as they can, as quickly as they can.
- Save lots of designs, and maybe even print them out to compare them to each other.

Approach:

More akin to an intermediate quilter since they are familiar with other forms of fiber arts. Will probably pick up quilting super easily.

User Case 5

Background: Amanda is a researcher at Rensselaer Polytechnic Institute studying the groundbreaking field of eTextiles. Specifically, Amanda is working on developing heated blankets for premature babies that would normally spend much of their early life in heated incubators. She's hoping to bring these infants and their mothers back together using heated blankets, which require repetitive patterns to make heating elements. She hopes to create normal looking designs to hide these elements in. She is familiar with quilting, but not the best at being creative.

Goals:

- Generate repetitive designs in which to hide circuit elements.
- Generate attractive designs to comfort patients.

Potential problems to look out for:

- Amanda might wish for more complicated designs, or more shapes to be offered.
- It would be cool for the generator to connect specific design elements together.
- Freemotion design generation could be an excellent asset for this specific application (using nichrome wire in the freemotion stitching).

Behavior:

- Appreciates the square based designs as grids are tileable and easy to use.

- Comes up with a handful of patterns for the research aspect, although may ever actually produce only one pattern.
- Would look for ways to edit the generator for her own specific goals if it was available as an open source project.

Approach:

She is an expert quilter so she would probably use this like a sketchbook for quilt designs. Could perhaps use colors to designate certain parts of a circuit rather than just for what they are. Thus, the color generation is secondary.

3.6 Technical Challenges

When approaching the goals of the project, some challenges were immediately identified. Technical challenges primarily stemmed from the goal: to create a soft, fabric controller and to communicate to and from the web tool. Although these challenges did not cause major roadblocks in the project, they were important in the planning and design phase, and will have an impact on the final deliverable.

Traditional components in controllers are designed to be hard, rugged, and designed to be handled aggressively and without constraint. But, this project aimed to make a controller that is both bendable and flexible. In order to accomplish this, most hardware that is used normally had to be substituted to find a better fit. For example, a hard, pushable button had to be substituted with conductive thread sewn into faux “button” contact pads that can detect when a user has touched it. This example extended into other aspects of the design. We had to find creative ways to incorporate electrical components into a soft, fabric medium. This technology is still new, as up until the mid 1990s electronics had not evolved to be small enough to be wearable (LOOMIA

2016). With newer technology, there is much less research information and data. Moreover, because wearable devices are few and far between, there are limited suppliers that sell components that are designed for fabric, which adds the limitations of cost and time.

Some specific issues that we ran into on this project with fabric, electrical components were with the slider and the buttons. Originally, we had intended to embroider a slider with conductive thread to measure the resistance based on the finger's location on the slider to determine the selected color. However, we ran into the issue of that with just a finger, there was too much variation in resistance to consistently and accurately measure the change in resistance. We were able to achieve more consistency by using the ground pin from the power supply in the lab so we thought that a grounded stylus might be an appropriate solution. However, after some research into soft potentiometer sliders, we found one that would work for our purpose. The slider we ended up using was the Soft Potentiometer Ribbon Slider from https://www.adafruit.com/ (Adafruit Industries 2022). In the final prototype we did find that this slider was not the appropriate size for the prototype, but it gave us more control and accuracy for our prototype. Additionally, we ran into a similar issue with our embroidered buttons. Using just our finger was not enough of a voltage change to register a touch with the microcontroller. After a bit of research, we found the MPR121 capacitive touch sensor breakout board which would allow us to more accurately read whether the buttons were being touched.

Furthermore, one of the biggest challenges discovered was interaction with the controller and web tool. Conventional controllers are designed as input-only devices, meaning the controller can send data to the device receiving the input, and the game, or tool, then uses real-world sensors to do something virtually. There are minor exceptions to this, as many controllers these days have "vibrating" motors built into the controllers to give a form of haptic feedback to the user. An example of this being the DualShock controller system from Sony for PlayStation game consoles. However, to satisfy the goals of this project, data needed to be sent back to the controller often as well as the controller sending input back to the web tool. Without the data sent to the controller, the device would be unable to know the current state of the tool, leading to a less intuitive and unhelpful controller. Normally, in video games this would not be too difficult to send and receive data from a controller as video games that use controllers are usually designed to have access to low-level system resources, and as such are usually written in C++, C#, or Java (Tyler 2021). This web tool was designed with a different approach; that is to run in any web browser on nearly any device. Web browsers are designed with much security and any code executed on a browser, in JavaScript, is at a much higher, sandboxed level than a standard videogame (Nadalin 2019). Thus, the issue is that there is no easy way to communicate with an external device connected to the computer through the web tool.

3.7 Project Timeline

This project timeline was planned to occur over the 2021-2022 WPI academic year through three 7-week academic quarters. This project was completed from A-term 2021 through C-term 2022 with different project aspects being completed in each term as detailed in Figure 3.2, Figure 3.3 and Figure 3.4.

In A-term, the project goals included idea generation, background research and initial design development. Initial development of project goals and ideas were developed early in the term, with background research on eTextiles, CodeCrafters, quilting design, and game controllers being conducted throughout the term to inform further system development. The second half of the term marked the beginning of design development, and these goals and their timelines are portrayed in Figure 3.2 below.

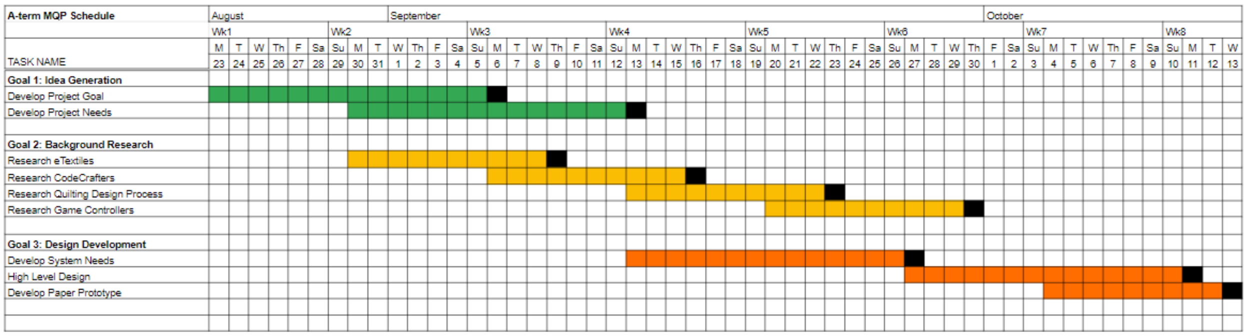

Figure 3.2: Gantt Chart of A-term Project Timeline

In B-term, project goals included continued development of the paper prototype, continued background research and report writing as well as initial system prototype development. A paper prototype to model user interaction was finalized and tested with human subjects to compile user feedback to inform further system development. Progress was continued with researching background information for the project and completion of sections of the project report. Finally, initial prototype design and building began later in the term, and these activities and timelines are outlined in Figure 3.3 below.

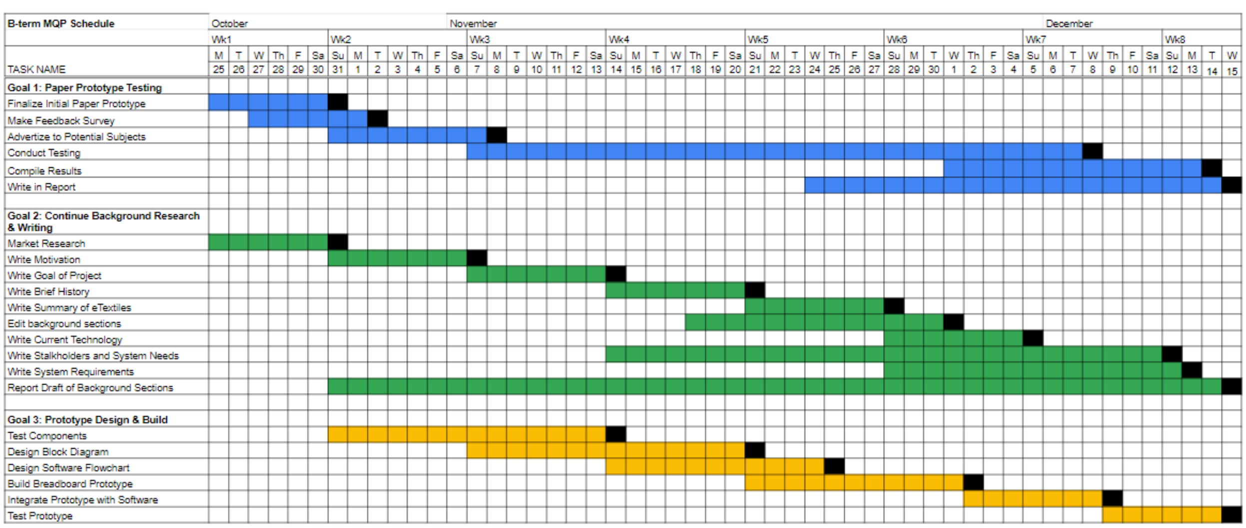

Figure 3.3: Gantt Chart of B-term Project Timeline

In C-term, final development of the system prototype occurred and the project report was completed and revised. This timeline is outlined in detail in Figure 3.4 below.

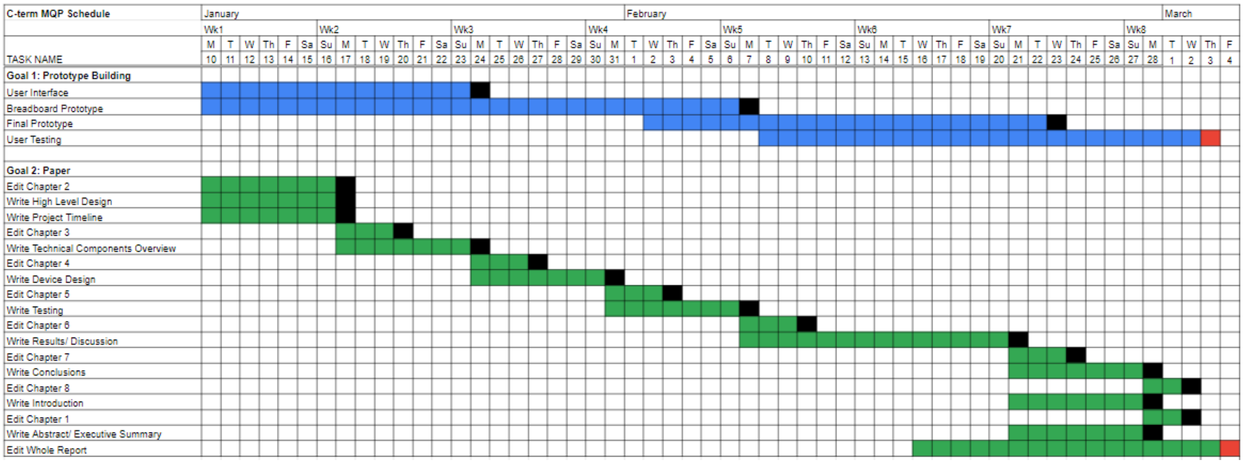

Figure 3.4: Gantt Chart of C-term Project Timeline

4.0 Technical Components Overview

In this chapter, we examine the available options for technical components to use in the construction of our initial prototype. We present some component trade-off studies we conducted to identify the best components for our system's requirements, which were established in previous chapters.

4.1 Component Trade-off Studies

For our design, we used specific hardware components to achieve the necessary functionality of each subsection of the system. The primary hardware components that needed to be evaluated were the display, conductive thread, microcontroller, and the battery. The display was needed to allow for visualization of patch designs on the fabric controller as well as providing visual user feedback. The conductive thread needed to be investigated for its use to create capacitive touch sensors as buttons. The microcontroller's purpose was to interpret the data from the controller and send the appropriate information to the web interface and vice versa so we needed to examine options for this critical component of the system. Finally, we needed to examine options for powering this system.

4.1.1 Display

As part of our design, we needed a way to display the user's patch design on the controller for optimal human computer interaction. For this module of the design, each display option was evaluated on the following specifications with no specific priority:

1. The display must be able to display multiple colors.
2. The display must be able to receive data from the microcontroller.

3. The display must be flexible to withstand the flexible, fabric nature of this system.
4. The display must be reasonably priced.
5. The display must have low current draw.

There are many effective methods to include color displays in projects. Our biggest limiting factor was that we would be using this display on fabric. Therefore, the most important requirement for our display was that it needed to be flexible. Table 4.1 shows the traits of each device and compares them to the requirements listed above. While the TFT LCD display had a better price and current draw than the LED Matrix, the LED matrix was flexible. Thus, we selected the NeoPixel RGB LED Matrix for our display.

Table 4.1: Display Trade-off Studies. The green, yellow and red coloring represents positive, neutral and negative traits of the device respectively.

	NeoPixel RGB LED Matrix	**OLED Breakout Board**	**TFT LCD Display**
	https://www.adafruit.com/product/2612	https://www.adafruit.com/product/1431	https://www.adafruit.com/product/358
Color	RGB	16-bit color	18-bit color
Microcontroller Compatibility	Compatible	Compatible	Compatible
Flexibility	Flexible	Not Flexible	Not Flexible
Price	$50	$40	$20
Power Consumption	1.2 A	30 mA	50 mA

4.1.2 Conductive Thread

One of the first design decisions we made was which conductive threads we were going to use and how we would use them. Much of the interface was intentionally designed to be composed of as much embroidery as possible for several reasons. Firstly, embroidery is a convenient way to decorate fabric items while also being durable and of good quality compared to other methods such as screen printing or fabric painting. In addition, most standard rigid or plastic electronic components are likely to break or distort if attached to a flexible substrate. Secondly, it was the best method of fabric decoration within the skills and resources available to our group. Finally, embroidery scales well from a business perspective while maintaining a similar quality, which would be optimal for stakeholders interested in seeing this project go to market.

4.1.2.1 Sewing

When used for sewing, the thread would be placed directly into the machine's bobbin or wound through the top of the machine to the needle. We used stainless steel thread for simplicity's sake, and while there are a variety of conductive threads on the market, stainless steel conductive thread is available to the masses and thus what we chose out of convenience. The conductive thread that was purchased was from Adafruit, which sells a variety of conductive textile products.

Figure 4.1: Spool of Conductive Thread (Wikimedia Commons 2016)

4.1.2.2 Embroidery

In order to create the embroidery we had to select a conductive embroidery thread to use as it is not interchangeable with normal thread. Madeira, an industry standard manufacturer and distributor of embroidery thread and products, sells two kinds of conductive embroidery thread, HC-12 and HC-40, with resistances of < 100 Ohm/m and <300 Ohm/m, respectively.

A sample stitch-out design for each of these threads was designed. This test also contained a sample touch button as well, as described below. We wound up going with HC-40 because it worked better within the constraints of our equipment and looked nicer, even though the HC-12 is less resistive.

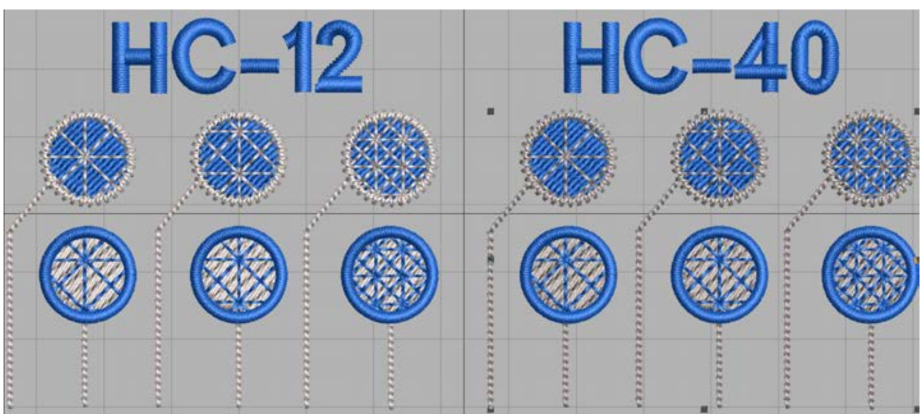

Figure 4.2: Pattern used to Test each Type of Conductive Thread

4.1.2.3 Button and Slider Design

All of the embroidery was designed in Wilcom Hatch, an embroidery digitizing software "for the hobbyist and home embroiderer" (Hatch Embroider 2022). Due to the size of the final project, compared to the limitations of the available equipment, our design had to be hooped three times. Industrial scale iterations should either be smaller or using a larger machine. In addition, the control panel was mistakenly placed on the wrong side of the tile placement grid. Unfortunately, due to time constraints, we were unable to remedy this mistake before further prototype building needed to happen, however, this did not significantly affect the user experience.

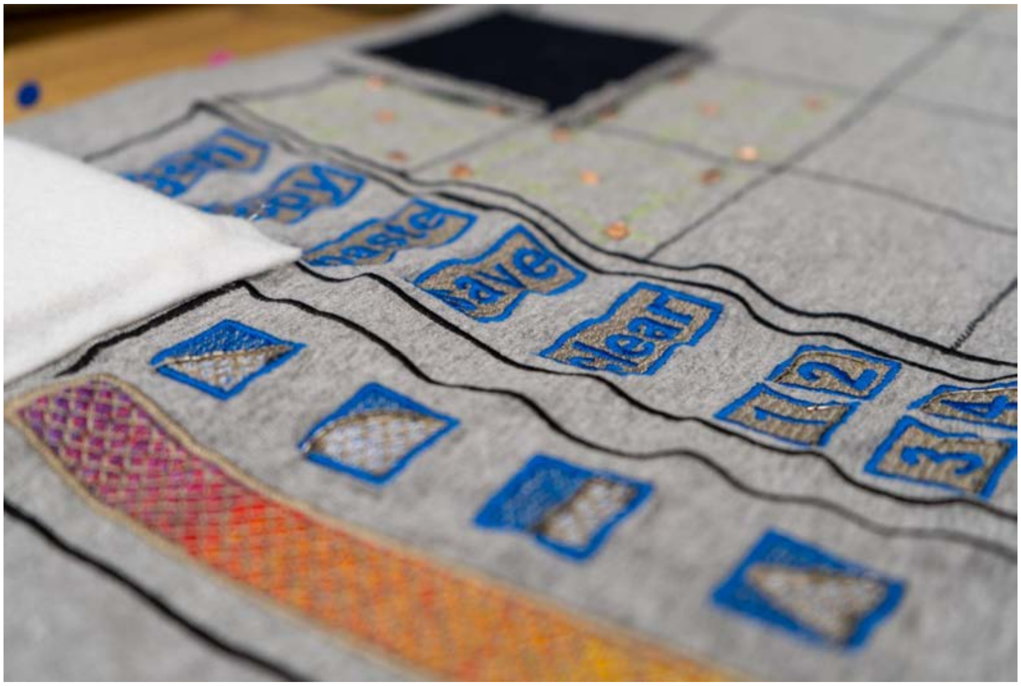

Figure 4.3: Closeup of Stitching on the Controller

Conductive thread is significantly more expensive than standard polyester embroidery thread. Thus a cost-effective solution minimizes the use of conductive thread. In a similar vein, an excess of stitches in general can increase time to stitch out, wear on the machine, and additional unnecessary cost in thread.

In order to experiment with minimalist design aesthetics, conservation of stitches (and thus thread, time to stitch, and wear on the machine), the buttons were designed to have a thin "mesh" of conductive thread over a colorful fill. This layering was surrounded by a thick "satin" stitch to further emphasize its color and further separate it from nearby buttons. This design approach was used throughout the design to create the designs shown in Figure 4.3.

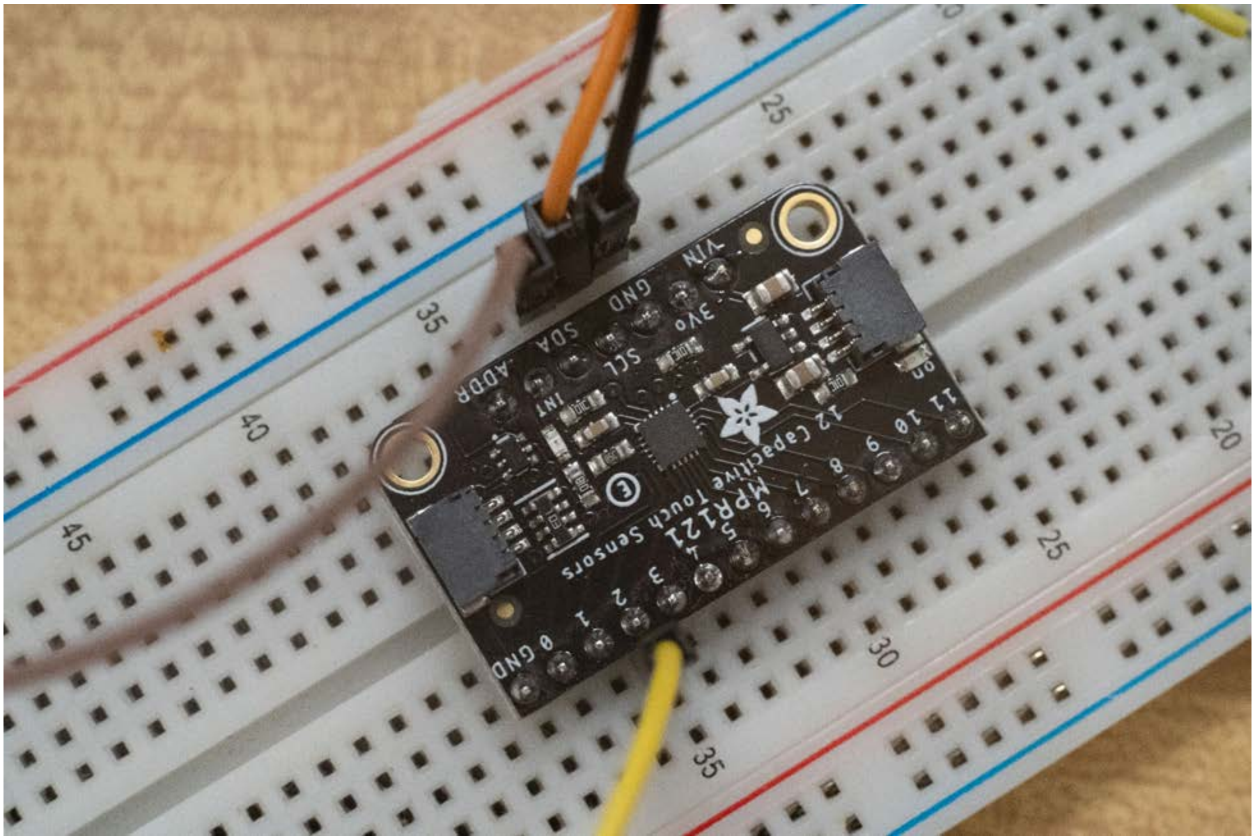

Figure 4.4: MPR121 Capacitive *Touch Sensor Breakout Board*

We originally intended to make a capacitive potentiometer fabricated in a style similar to the buttons, but in the shape of a long rectangle. A “mesh” of conductive thread embroidered in an aesthetically pleasing pattern was stitched over top of a color gradient to designate a brightness slider. Unfortunately, we were not able to get this design to work as we had expected it to well and needed to pivot.

Fortunately, flexible potentiometers are readily available. We purchased two similar in size to the sliders in our half scale design and attached it to the final product using embroidery thread. Each lead on the potentiometer was connected to their designated Arduino pins while the main surface was glued to the back of the device.

Figure 4.5: Closeup of Fabric Slider and Buttons

4.1.3 Microcontroller

Communication between the controller and the web tool is critical in the design of the controller. The microcontroller chosen for this task is an Arduino Leonardo. The Leonardo was chosen for the specific feature of it that many other microcontrollers lack. The microcontroller, which is based on the ATMega32u4, has built-in USB communications, which make it compatible with the WebUSB specification, with some minor firmware changes.

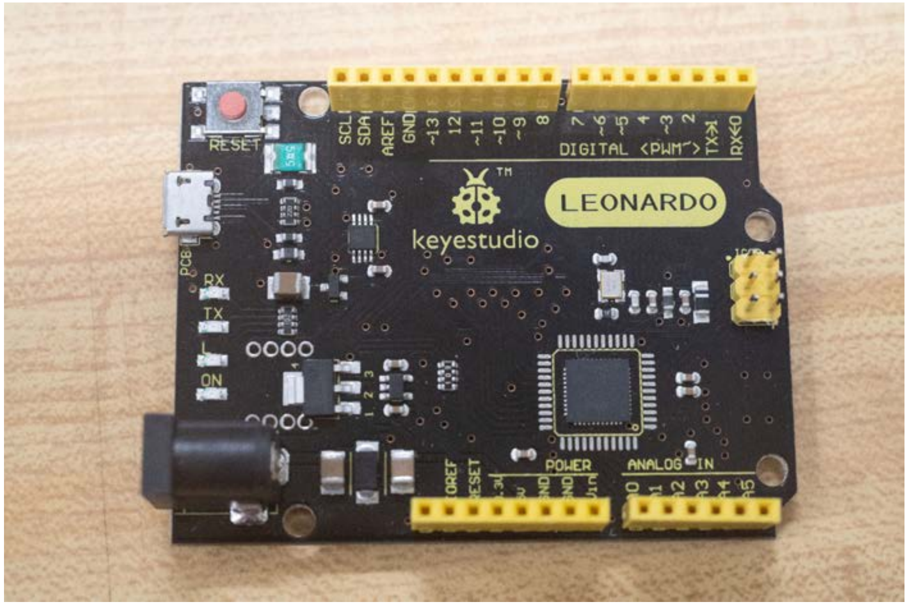

Figure 4.6: Arduino Leonardo used in Device Prototyping

Another option for a microcontroller to use in this device was the ATMega328p. Compared to the ATMega32u4, the ATMega328p is the main microcontroller used in the Arduino Uno, the original Arduino development platform. It is a decent competitor because the entire Arduino ecosystem was developed based on this specific chip, so it is the default chip used in most hobbyist projects and support for it is easy to come by. However, this microcontroller was ultimately not used because as the web interface was developed, it was discovered that the ATMega328p did not have enough memory to handle some of the communication methods used in this project. Furthermore, it does not have a built-in USB chip that supports the WebUSB standard thus, this microcontroller would not work for this application.

Another alternative was the ESP-8266. This microcontroller is an excellent chip because of its attractive price point and built-in Wifi abilities. Several development boards for it are available online, and its sister microcontroller, the ESP-32, adds bluetooth communication as well.

However, neither of these chips were chosen because our browser communication ultimately was done over serial rather than Wifi, diminishing some of its appeal. It was also discovered upon experimentation that the development board we had purchased did not have enough pins with which to read analog values, causing us to decide against it in order to continue development at a good pace. Furthermore, this microcontroller also did not have the required built-in USB communication, thus also disqualifying it for not being compatible with the WebUSB standard.

4.1.4 Power

This device required a method to power the components of our design while also maintaining the portability and flexibility of the device. While power from the computer though the microcontroller was enough to power most components, our LED display needed more power capabilities than the microcontroller is capable of providing. When evaluating power supplies, we considered the following needs:

1. The power supply must be able to power the LED display. Therefore, the power supply must have a nominal voltage of 3.7V.
2. The power supply must be lightweight.
3. The power supply must be rechargeable or easily replaceable.
4. The power supply must be reasonably priced.
5. The power supply must be able to power the device for a few hours at a time. Therefore, the power supply should have a capacity of at least 2500mAh.

In order to meet the requirements of this system being portable and easy to use, it was important for the power supply to be lightweight and easily recharged or replaced. We also wanted to keep the device reasonably priced, thus a cheaper battery was ideal. For the components that we used in this system, we needed a power supply that could supply the necessary 3.7V nominal voltage. And in order to power our device for a few hours of use, we needed a power supply with at least 2500mAh. We determined this value through the typical current draw of our components coming to about 1.2 A. Therefore, 2500mAh would keep our device running for at least two hours.

Table 4.2 below outlines the trade-off studies of a few power supply options. Based on the factors stated above and the traits of the power supply options outlined in the table, we settled on the Adafruit Lithium Polymer Ion Battery for our device. This battery offered the best portability and price with an adequate capacity while also being specifically designed for eTextile applications. The latter point was appealing because of its potential for smooth integration and useful documentation within our own product.

Table 4.2: Power Supply Trade-off Studies. The green, yellow and red coloring represents positive, neutral and negative traits of the device respectively.

	Adafruit Lithium Polymer Ion Battery	Jauch Quartz Lithium Polymer Ion Battery	Lithium Ion Battery
	https://www.adafruit.com/product/328	https://www.digikey.com/en/products/detail/jauch-quartz/LP906090JH-PCM-2-WIRES-70MM/9560999	https://www.digikey.com/en/products/detail/adafruit-industries-llc/353/5054549
Nominal Voltage	3.7V	3.7V	3.7V
Weight	43g	102g	155g
Rechargeable/ Replaceable	Rechargeable	Rechargeable	Rechargeable
Price	$15	$33	$24.50
Capacity	2500mAh	6Ah	6.27Ah

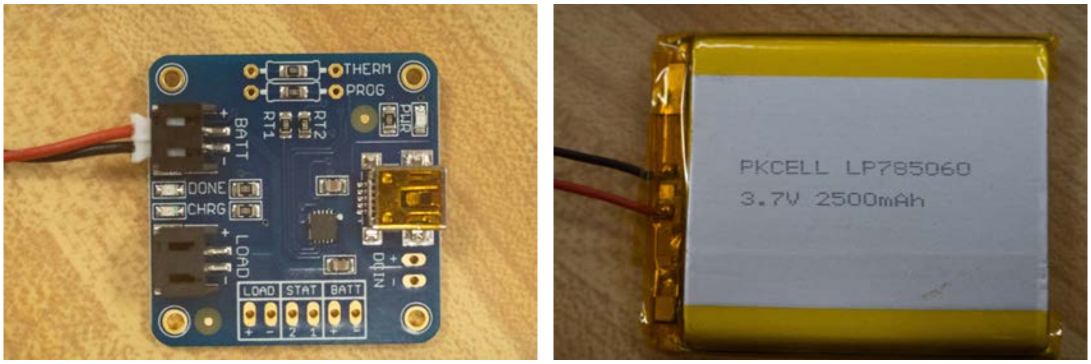

Figure 4.7: Adafruit Lithium Polymer Ion Battery and Charger

5.0 Device Design

In this chapter, we will walk through the final device design of the system. We will start by explaining the hardware design followed by the design of the software. Then, we will explain how we were able to interface between the device's hardware and software.

The following figures show the final prototype of the design mat. The final prototype was built using a combination of sewing, hot glue and electrical tape to secure our electrical components to our embroidered controller mat.

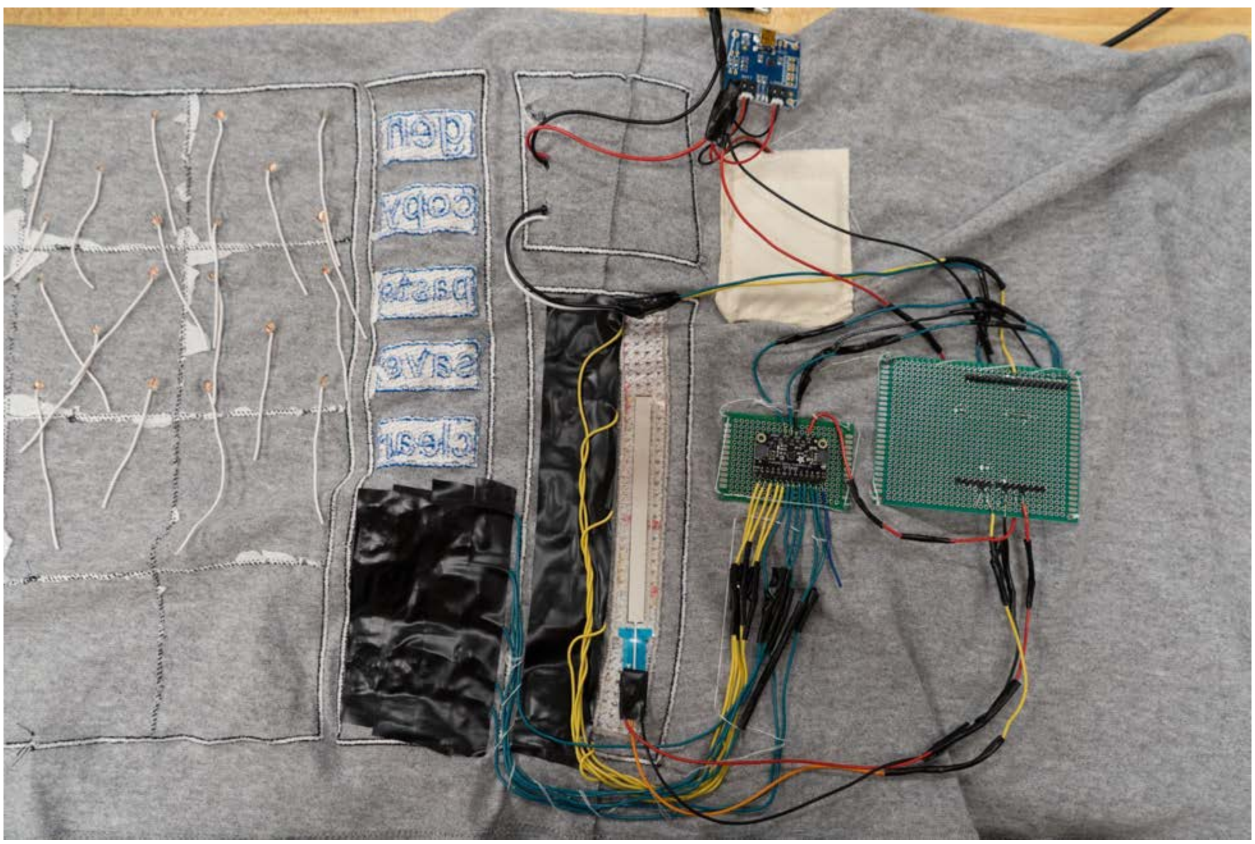

Figure 5.1: Final Controller Hardware (Without Arduino Microcontroller)

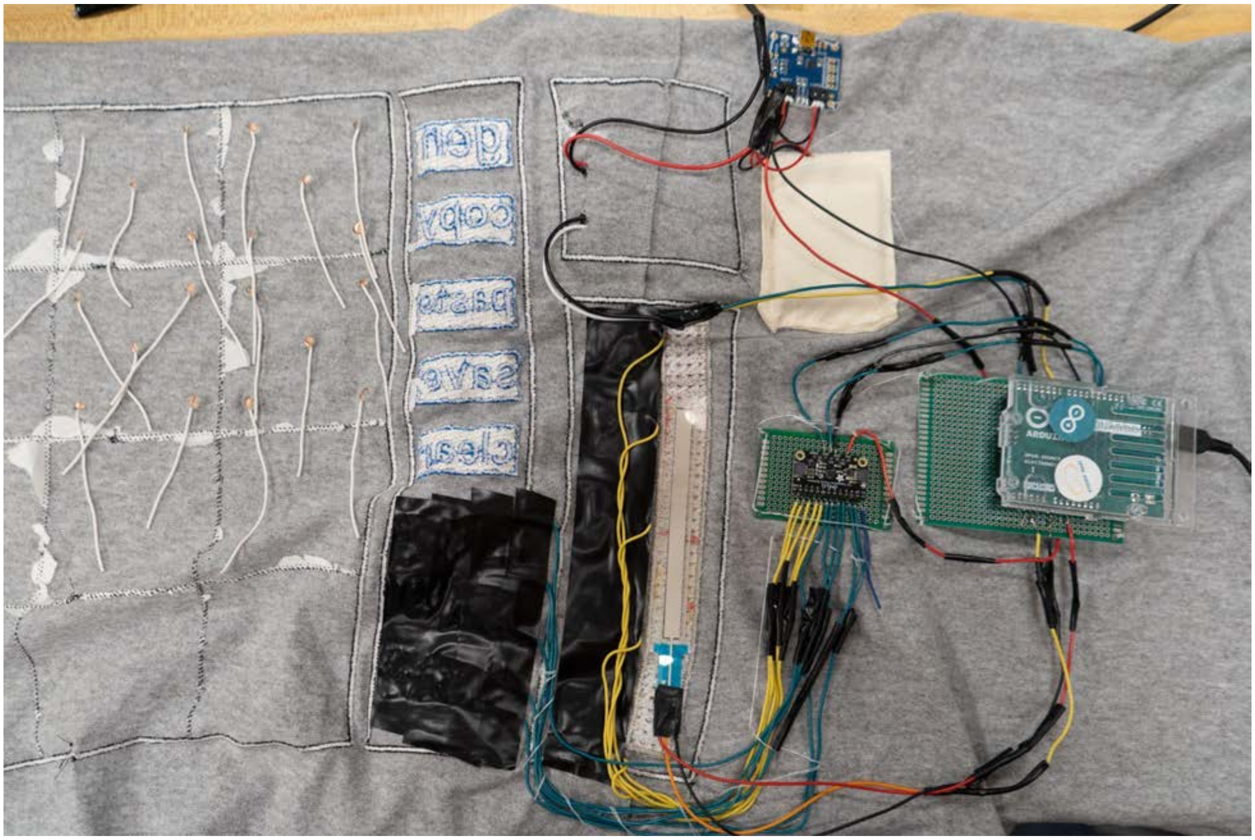

Figure 5.2: Final Controller Hardware (With Arduino Microcontroller)

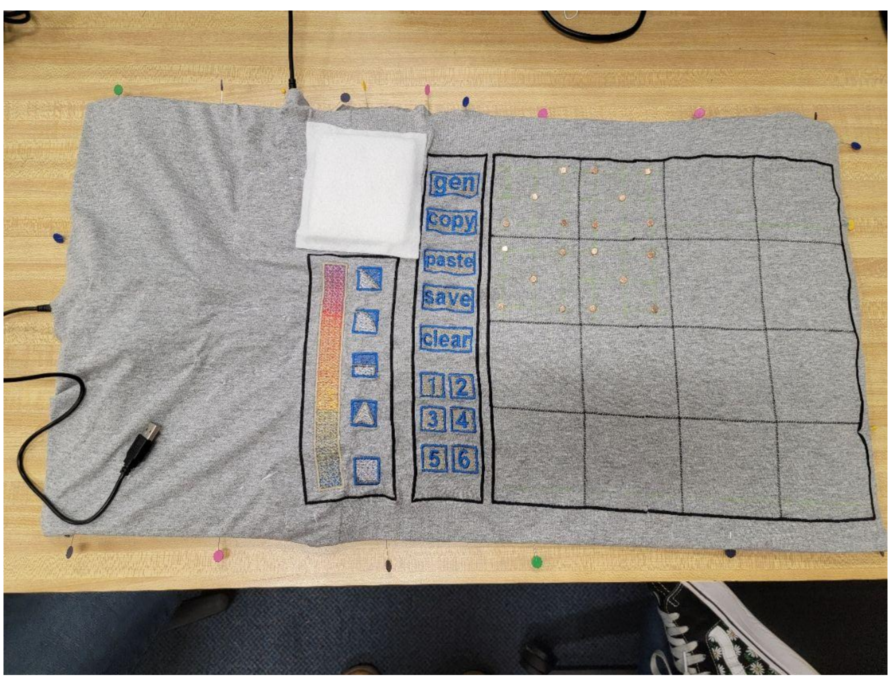

Figure 5.3: Final Device Prototype

5.1 Hardware Design

This device was prototyped using an Arduino, a popular development board for the ATmega328p microcontroller due to its large community of support and simplified programming language. To accommodate for the large number of inputs our design requires, the Texas Instruments SN74HC158N multiplexer is used to control the flow of data from each sensor. The multiplexer receives a specific binary sequence from the microcontroller, allowing this chip to select from 2 of 8 streams of data, effectively generating more space for inputs.

The face of the device was embroidered using an array of conductive and non-conductive embroidery thread to designate key areas of the device's design as seen in Figure 5.4 and Figure 5.5. Individual buttons such as the number pad and patch select buttons were embroidered in conductive thread, creating a surface on which to sense touch. The MPR121 breakout board is used to detect touch, which reports back to the ATmega328p. The non-conductive thread was used to designate key areas such as the tile grid and add aesthetic appeal. The circuit design for these buttons can be seen in Figure 5.6.

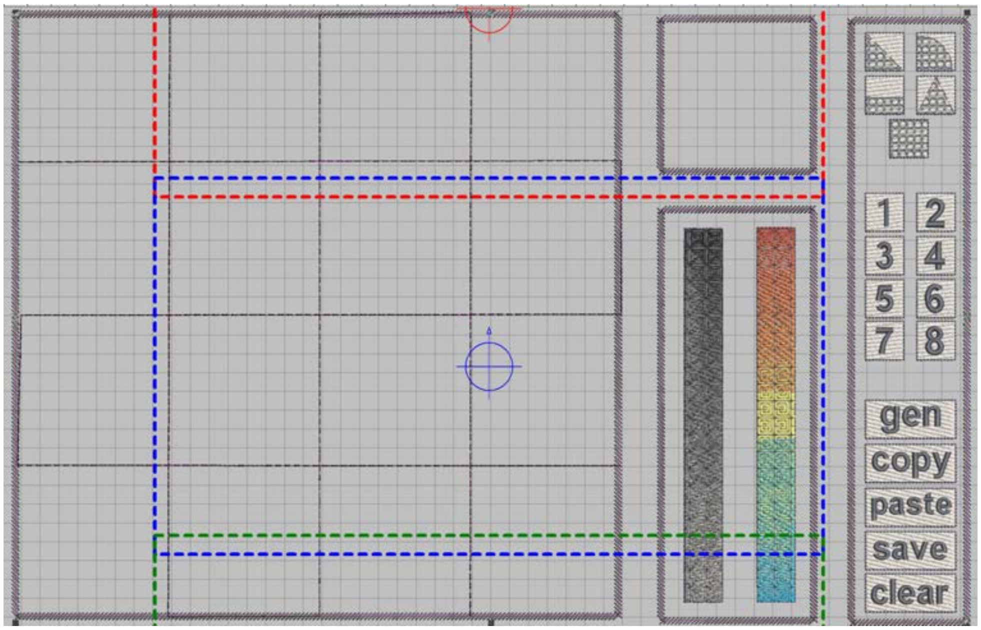

Figure 5.4: First Draft of Embroidery Pattern for Device

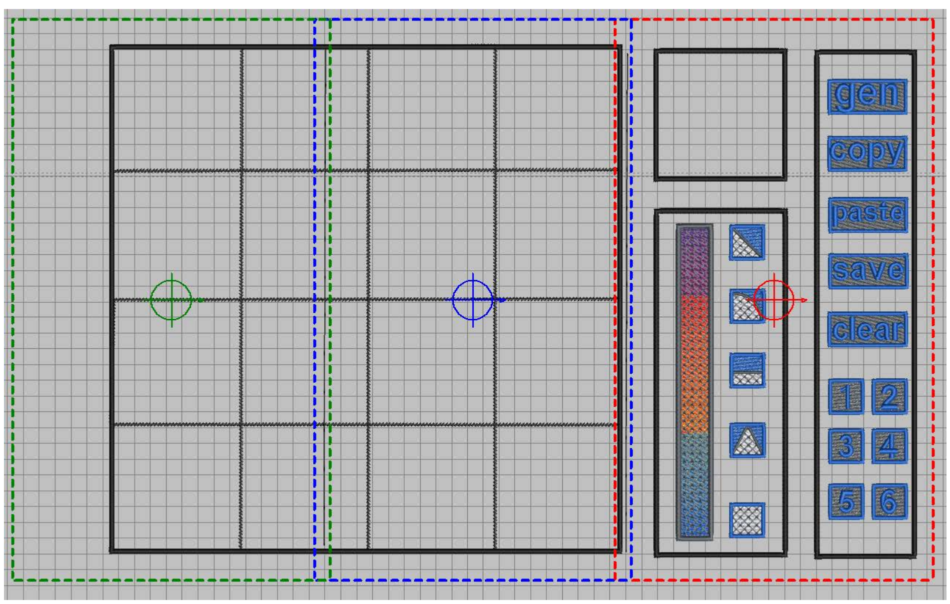

Figure 5.5: Final Design of Embroidery Pattern for Device

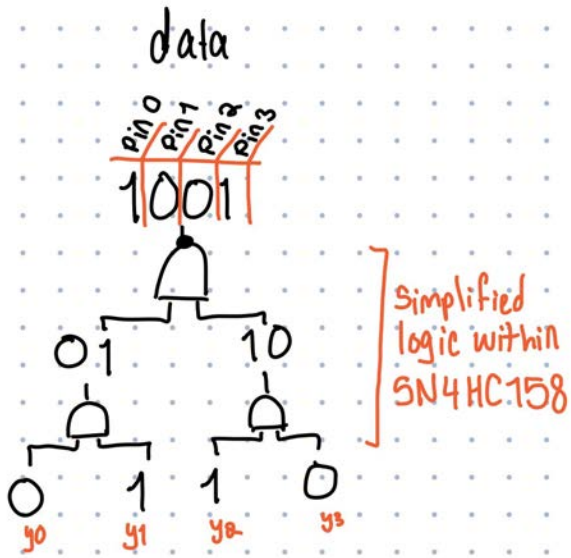

Figure 5.6: Circuit Diagram for Embroidered Buttons

The tiles use a clever arrangement of magnets to communicate what orientation they are in. Five 2mm magnets are encased in copper foil and attached to a square of durable fabric. Four of these magnets are connected asymmetrically, leaving one magnet as a stabilizing placeholder as seen in Figures 5.7 and 5.8. When a tile is placed within a square on the device's tile grid, these magnets ground a handful of probes (which also are attached to 2mm magnets) tied weakly to a voltage source in such a way that the ATmega328p can sense the tile's orientation. Figure 5.9 shows a diagram of how the multiplexers were used to connect this circuitry to successfully identify tile placement and orientation.

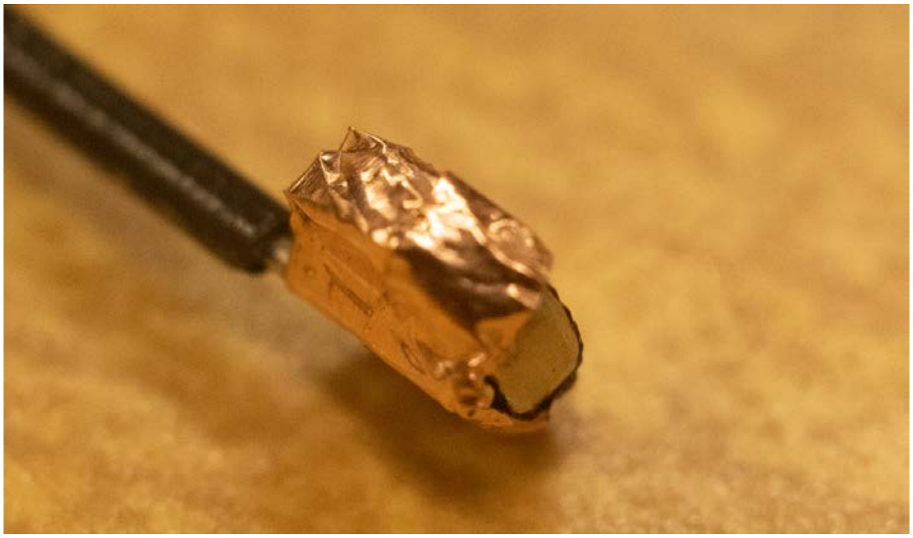

Figure 5.7: Magnet Encased in Copper Foil and Attached to Wire

Figure 5.8: Final Tile Design

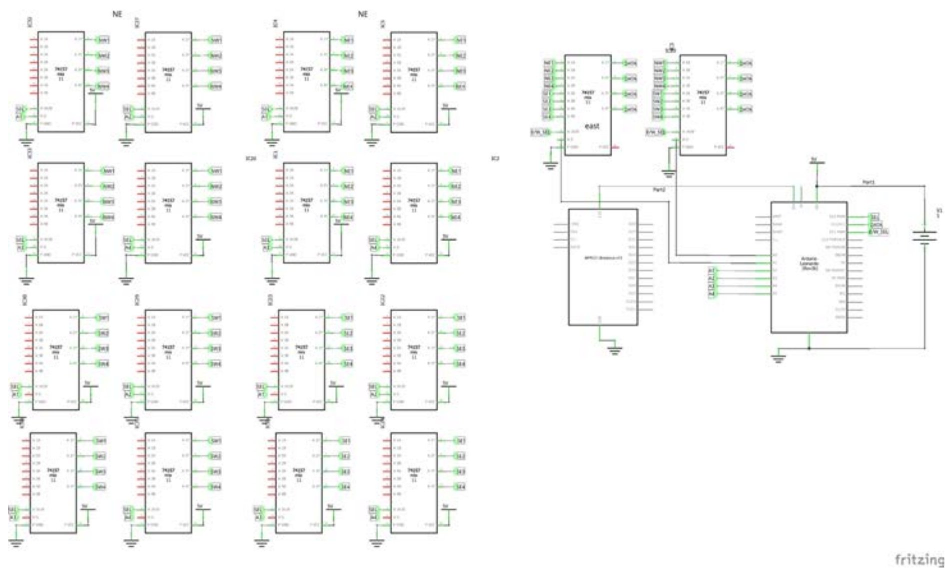

Figure 5.9: Prototype Schematic of the Tile Sensing System

To provide power and data to our LED matrix so that it could successfully be used to display the user's design on the controller, we used the circuit diagram found in Figure 5.10. We used a capacitor across the battery supply and a resistor between the data pin and ground as protection elements to keep the LED matrix from being damaged if there were any power spikes. Figure 5.8 shows initial breadboard prototyping of this circuitry.

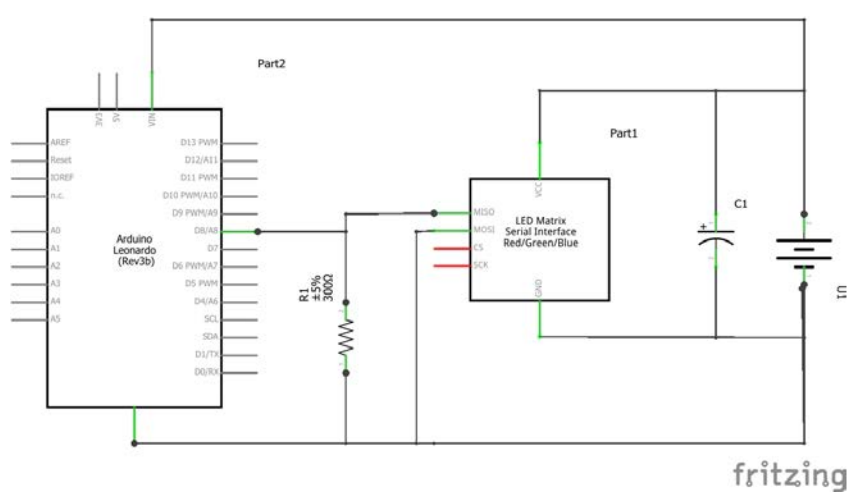

Figure 5.10: Schematic for the LED Matrix

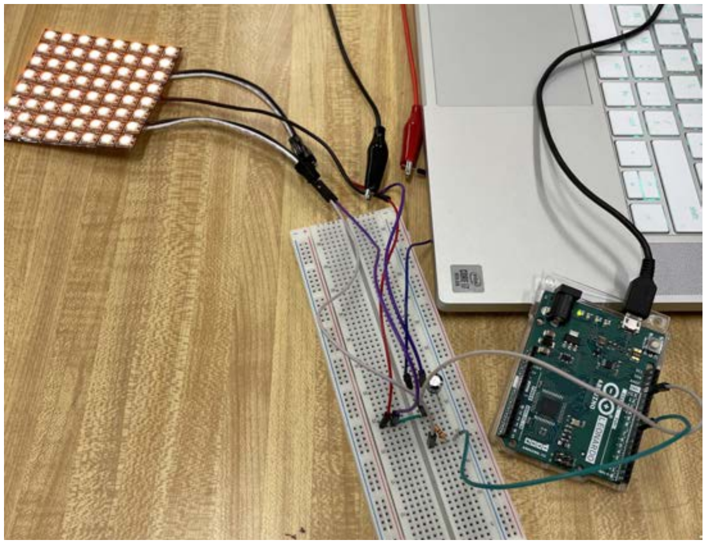

Figure 5.11: Breadboard Prototyping of the LED Matrix

In order to control color selection on the controller, we used a soft potentiometer slider that communicated the location of a finger to the microcontroller through variable resistance. Through testing we found that for a few seconds after the finger was removed, there was some variation in the color that was being communicated to the microcontroller. Rather than using a complicated software fix to this problem, we were able to connect a capacitor across the data and ground pins of the slider to hold the value after the finger was removed. The circuit diagram for this slider is shown in Figure 5.12 and the initial breadboard prototyping of this segment of the device is shown in Figure 5.13.

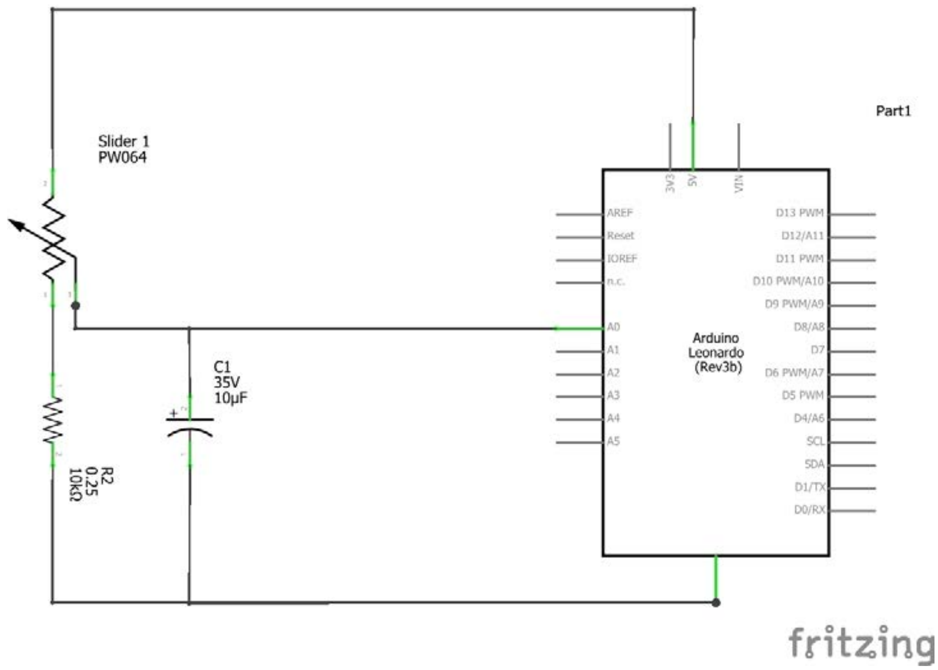

Figure 5.12: Schematic for the Color Slider

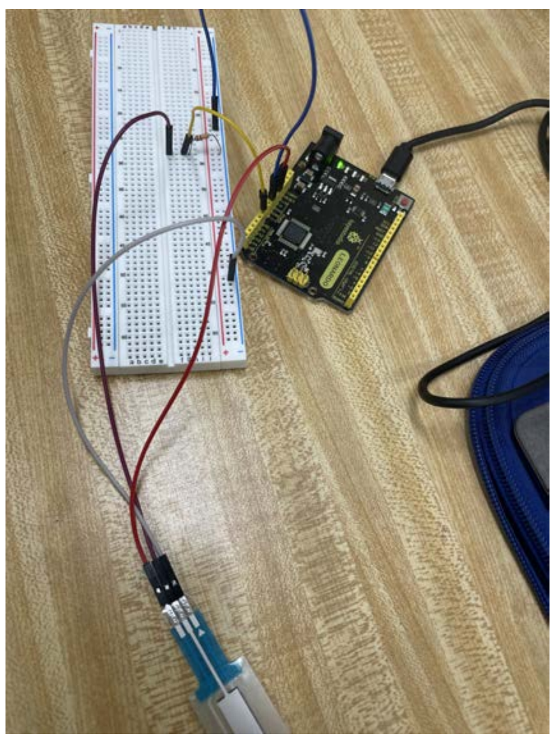

Figure 5.13: Breadboard Prototyping of the Slider

For our final prototype, we soldered this circuitry on a PCB design board as seen in Figure 5.14 below to attach to the microcontroller. We used a combination of sewing, hot glue and electrical tape to secure our electrical components to our embroidered controller mat.

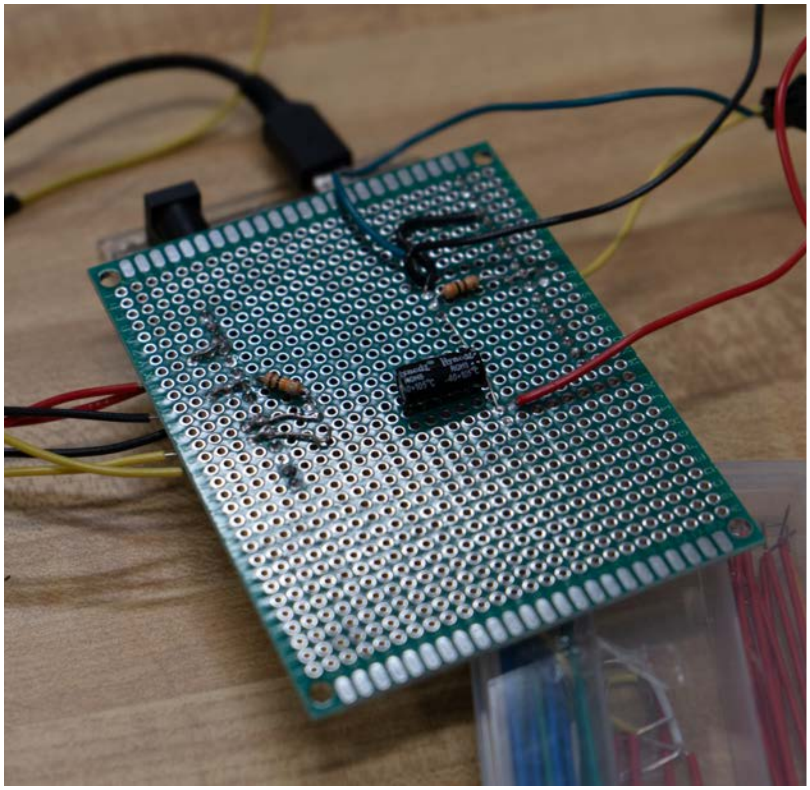

Figure 5.14: Soldered PCB Shield for Microcontroller

5.2 Software Design

This section outlines the design for the software component of this device. This covers the development of the website or web interface and the required software security for this prototype.

5.2.1 Website/Web Interface

The website for this project was forked from the original Code Crafters project. With the code having been written by different authors, the first thing that needed to be done was understand what they did and if additions and changes were to be made. The code was written for the web using Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript, using Node.js as the backend server. These languages are the backbone of every modern website, making it the best choice to support our project with. Having HTML being the defining code of what will appear on the site, the CSS being the styles of the elements on the page, and the JavaScript being the interactive code that can dynamically change any aspect of the page, as well as dynamically interact with remote servers and communicate with the Arduino microcontroller over the WebUSB standard.

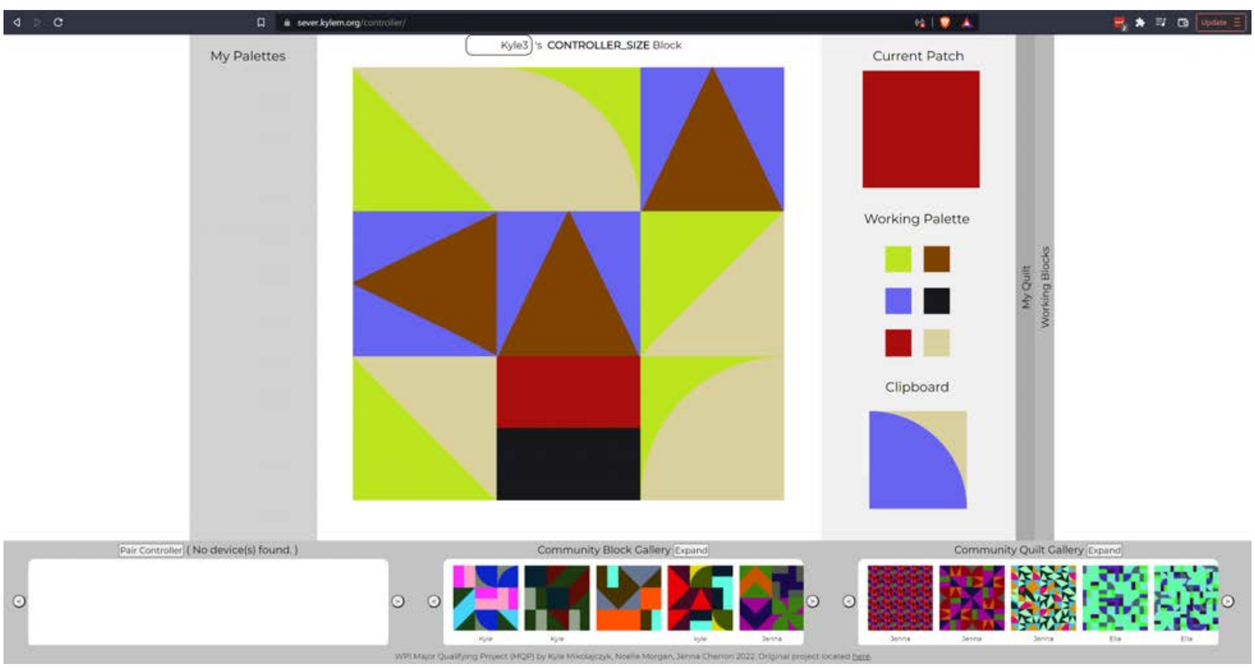

Figure 5.15: UI of Web Interface

The original writers of the software used some JavaScript libraries to help them, specifically the Google Firebase library. This allowed them to connect to a Google Firebase database easily, which is where the user's designs are saved and shared from. The first step done when making our own modifications to the website was to create a new Firebase database so that our testing data did not conflict with the current, live users. Once this was done, the next step was to modify the UI to fit the needs of this project. One of the biggest changes needed was the removal and simplification of the UI as most of the UI's functionality was being moved to the controller. The simplest solution was to hide the elements that were no longer needed by setting the "display" type to "none", or adding the "hidden" attribute to the element. The reason for hiding the buttons and elements was that these elements were still needed on the controller, and the quickest way to get the controller working for the program was to have the button press on the controller call the press event on the now hidden element within the website. This was not difficult, but the following modification of the style of the UI through the CSS took a long time as significant reorganization of the file was necessary, both because elements were now missing visually and to better make the UI match the physical controller that was being designed.

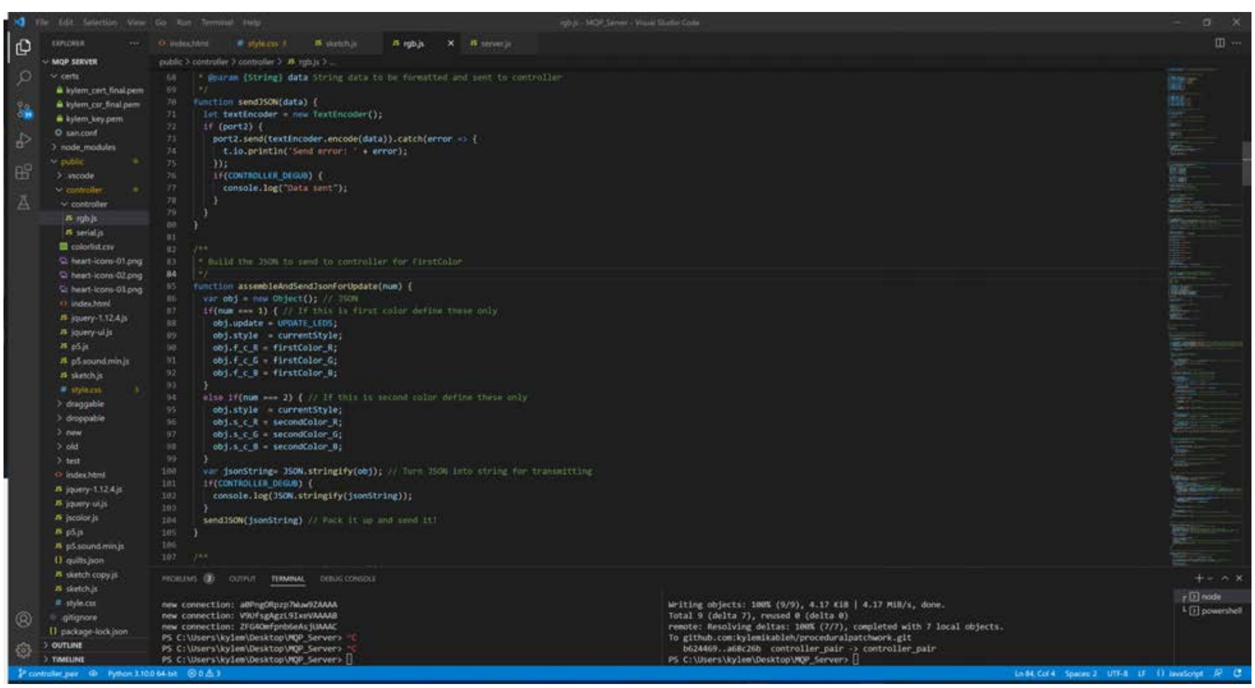

Figure 5.16: JavaScript Controller Code in Visual Studio Code Editor

The next, most important code that needed to be added was the JavaScript code that allowed for interaction with the Arduino through WebUSB and Websockets. There is an open source project, called WebUSB Arduino, which is a small library that allows for interaction between Arduino microcontrollers that have a USB chip on them (WebUSB Arduino 2022). Taking the example code for changing a physical LED from a website, it was overhauled to add various functions that check for JSON received from the microcontroller as well as sending JSON to the microcontroller. The JavaScript handling the communication is located in a separate file from the main site script to make organization easier. Functions in both files can call each other's functions and variables due to the way JavaScript shares references.

When certain actions are done on the controller, a JSON update will be sent over serial, triggering an event. On the trigger of the event, the string is serialized into a JSON, allowing us to read the key-value pairs. An IF statement checks the “action” type that was submitted, and calls the proper function that handles the “action”, looks for additional keys associated with that “action”, and updates the UI when necessary. If the “action” changes something that also needs to be reflected on the controller, an update will be sent back to the controller.

Furthermore, for sending updates to the controller, a similar process was followed. Anything on the UI that could cause an update for the controller had event handlers registered on a specific action, such as a button press or drag start/end event. When one of these occurred the data about the event was sent over to the controller handling file. From here data such as what tile was updated and the style and colors were placed in a JSON and sent over to the controller.

5.2.2 Required Software Security

When the user first loads the page, all the functionality of the original site is loaded. Once all the necessary data has loaded, the new controller code is added to the site. The user is presented with a “Connect Controller” or “Pair Controller” button, which when selected brings up the browser’s built-in WebUSB permission dialogue. Security is high on most modern browsers and the controller cannot automatically be paired for the first time without any initial “user interaction” such as a button press. After the permission is given, the events that are designed to function after serial is received starts listening for an event to be triggered.

Another important security requirement when using the WebUSB standard is the requirement of an encrypted connection to the web server. The standard states that the WebUSB library will not be usable unless the connection is guaranteed to be secure, to prevent any abuse by hackers (Beaufort 2022). The path to achieving this standard involved establishing a connection through the Transport Layer Security (TLS), which allows users to connect to the web server over Hypertext Transfer Protocol Secure (HTTPS) instead of the original, non-secure standard of Hypertext Transfer Protocol (HTTP).

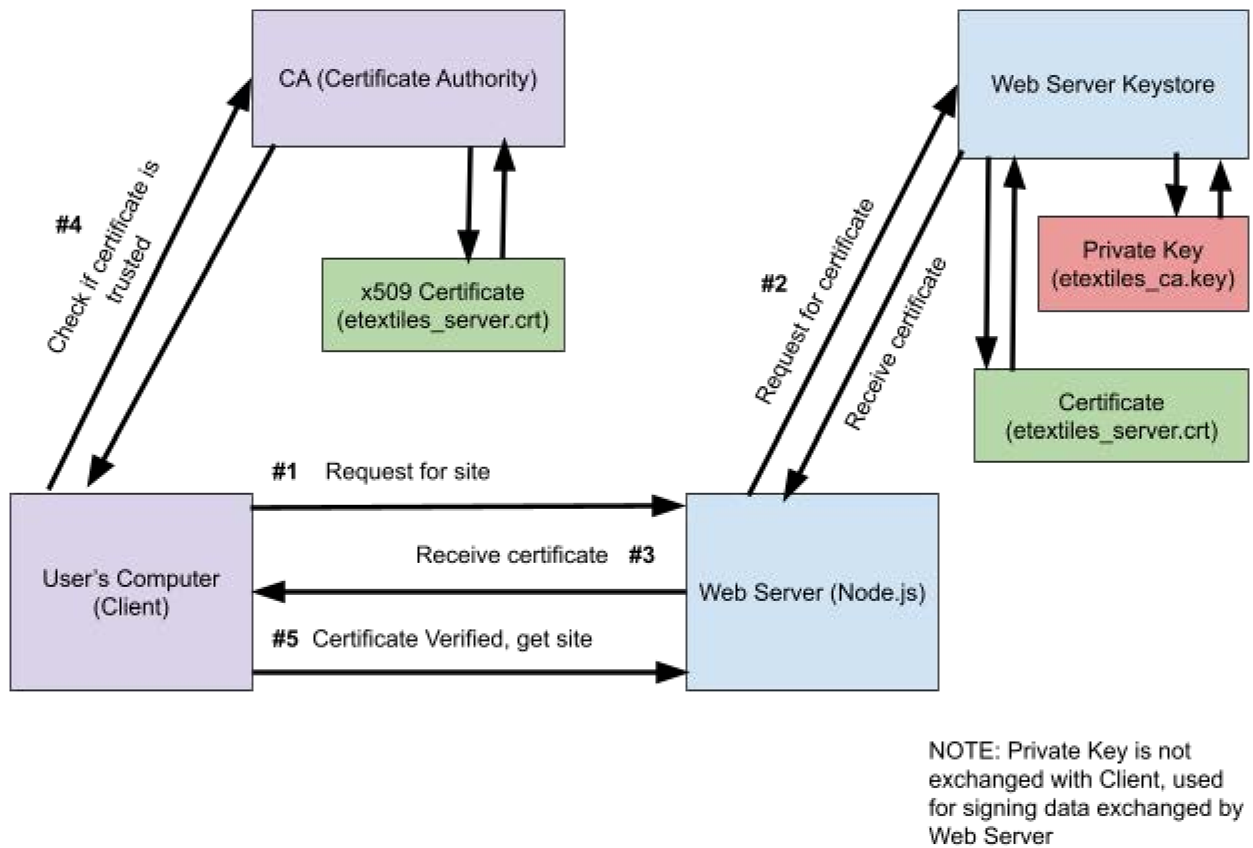

Figure 5.17: Diagram of TLS Handshake

Security over the internet over the past several years has improved and expanded exponentially, resulting in nearly every mainstream website now running over HTTPS. The TLS protocol works by encrypting the data being sent to and from the server, preventing hackers from observing the data being sent, and more importantly preventing tampering with the data being transmitted. For this connection to be created for the site for testing, a private key and public key needed to be created.

TLS certificates work by a user's browser trusting a signing authority that signed the TLS certificate for the site. Usually these are very large companies that are trusted by web browsers, and they can issue a server owner a key, either for free or for a fee. Due to the needs of the testing environment, the approach of making our own certificate signing authority (CSA) was decided as useful for this environment, but in a future released version of this project, a key from a well-known signing authority would be necessary.

Once our CSA was created with a private key and public key (using the ssh-keygen and openssl linux packages), the signing authority had to be trusted by the development computers connecting to the server, so that the TLS handshake would be trusted. Creating cypher keys is usually difficult because the standard instructions are widely available, but recent changes to the security standard required more configuration than browsers used to expect, specifically the SAN (Subject Alt Name) parameter is now required.

First, the Certificate Authority (CA) private key was needed to be generated, with the `openssl genrsa -out etextiles_ca.key 2048` command. This command generates an openssl CSA private key with 2048 being the size in bytes of the key. Once the CSA private key is generated, the CSA x509 certificate can be generated, which is a certificate that a user's browser can recognise and trust. The command that was used is `openssl req -x509 -new -nodes -key etextiles_ca.key -subj "/CN=Kyle Mikolajczyk/C=US/L=MA" -days 365 -out etextiles_ca.crt`. The important part of this command is that the "-days" parameter was the length before the expiration of the certificate; in this case the parameter was 365 days. Also, the "-subj" parameter specifies details about the CSA, which can be the name of the signer, the country code, and the location. Next, the TLS private key needed to be created with `openssl genrsa -out etextiles_server.key 2048`. For the Certificate Signing Request (CSR), a configuration file is needed to have all the proper identification information in the certificate for the browser to be satisfied with security requirements.

```
[ req ]
default_bits       = 2048
default_keyfile    = etextiles_ca.crt.pem #name of the keyfile
distinguished_name = req_distinguished_name
req_extensions     = v3_req

[ req_distinguished_name ]
countryName                 = US
countryName_default         = US
stateOrProvinceName         = MA
stateOrProvinceName_default = MA
localityName                = Worcester
localityName_default        = Worcester
organizationName            = Kyle Mikolajczyk
organizationName_default    = Kyle Mikolajczyk
commonName                  = sever.kylem.org
commonName_max              = 64

[ v3_req ]
keyUsage = keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names
```

```
[alt_names]
DNS.1    = sever.kylem.org
DNS.2    = sever.kylem.com
DNS.3    = localhost
```

Figure 5.18: csr.conf

The important parts of this configuration is that the "alt_names" include the different Domain Name System (DNS) addresses that this certificate can be used on. For this project, the testing domain was either "localhost" (for use on a local computer) or the remote server domain of https://sever.kylem.org. Moreover, using this configuration can be done with the command `openssl req -new -key etextiles_server.key -out etextiles_server.csr -config csr.conf`. Lastly, putting everything together with the following command makes the required CSA certificate; after which the key file and public key file can be utilized by the Node.js web server: `openssl x509 -req -in etextiles_server.csr -CA etextiles_ca.crt -CAkey etextiles_ca.key -CAcreateserial -out etextiles_server.crt -days 365 -extfile csr.conf`. With all this now accomplished, and when the user has imported the certificate onto their own computer, the use of the controller remotely on the development server is possible.

5.3 Hardware and Software Integration

The following section describes the process for integrating the hardware design from Chapter 5.1 with the software design from Chapter 5.2. First, we present how we accomplished communication between the controller and the web interface. Then, we present the Arduino code used to control the microcontroller and in turn communicate between the hardware and software.

5.3.1 Communication Between Controller and Website

The only way to achieve a controller which can both send and receive data from the website was through the use of the WebUSB standard that was recently introduced to most mainstream web browsers. This standard makes it possible to send and receive data to USB devices running with the USB 2.1 or higher standard, along with the proper header and description data. However, the current Arduino Leonardo needed some software modifications through a library for web browsers to recognise it on all popular operating systems. The open source library ArduinoUSB was found to provide the necessary modifications, and with the change of the USB version to 2.1, and the use of this library, the Arduino was able to be recognised by web browsers (WebUSB Arduino 2022).

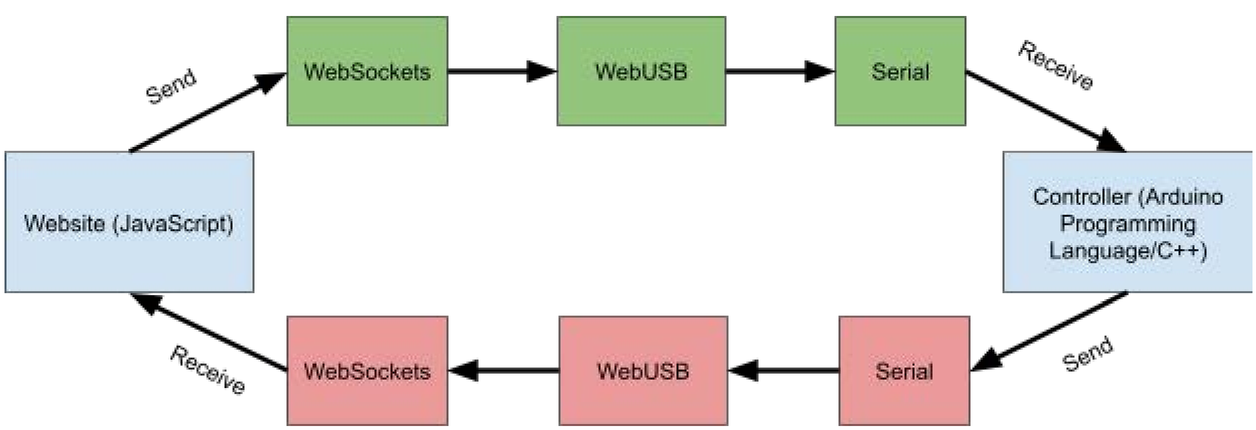

Figure 5.19: Diagram of Controller-Website Communications

Once the connection was established using the library for both the Arduino and the JavaScript browser, data had to be received on the Arduino through the serial connection. The ArduinoUSB serial connection is established in the setup function, and from there data is communicated via serial on the Arduino, and WebSockets on the web browser.

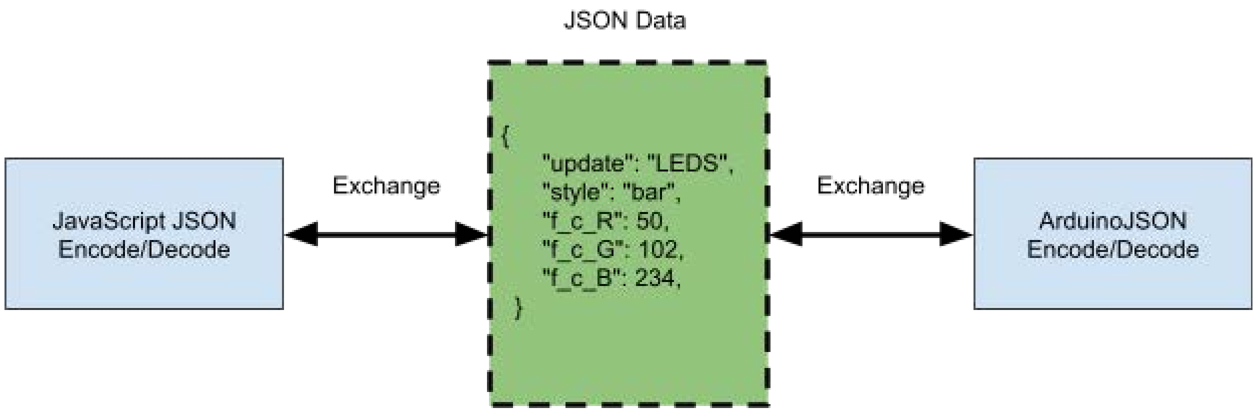

Figure 5.20: Diagram of JSON Data Exchange

The next challenge that needed to be resolved was the format of data that needed to be exchanged. With much thought, the simplest and most standard approach that was decided upon was to send JSON files to and from each side. For the JavaScript website side, JSON encoding and decoding is built into the language, making the data exchange easy. However, the Arduino Programming Language does not have a built-in JSON encoder or decoder, thus a library was needed for the Arduino; the “ArduinoJSON” library (Blanchon 2021). Also, because every character in the JSON takes up a byte of memory within the Arduino’s limited capacity, reducing the size of each key was critical to prevent running out of memory during transmissions or deserialization on the microcontroller. Once this library was included with the program, the Arduino became capable of sending and receiving data in real time.

5.2.2 Arduino

When designing the project, one of the main objectives was to keep the cost of the project to a minimum. The microcontroller that was capable of using the WebUSB standard that was chosen for this project was the Arduino Leonardo, among many other of its benefits towards our project. One of the critical downsides with this device was that the program space memory (Static Random-Access Memory, or SRAM) was limited to only 2.5Kb of SRAM, which significantly limited the approach to writing code for the project. The design of the code for the controller had to take the limited memory into account, requiring plenty of planning when determining the data types so as to not go over the maximum amount of SRAM being used at any time during the program's lifecycle. Furthermore, the use of libraries had to be limited as well because of these same limits, which was particularly challenging because the required libraries took up over 80% of the program space, leaving a minimal amount for the controller code on the microcontroller.

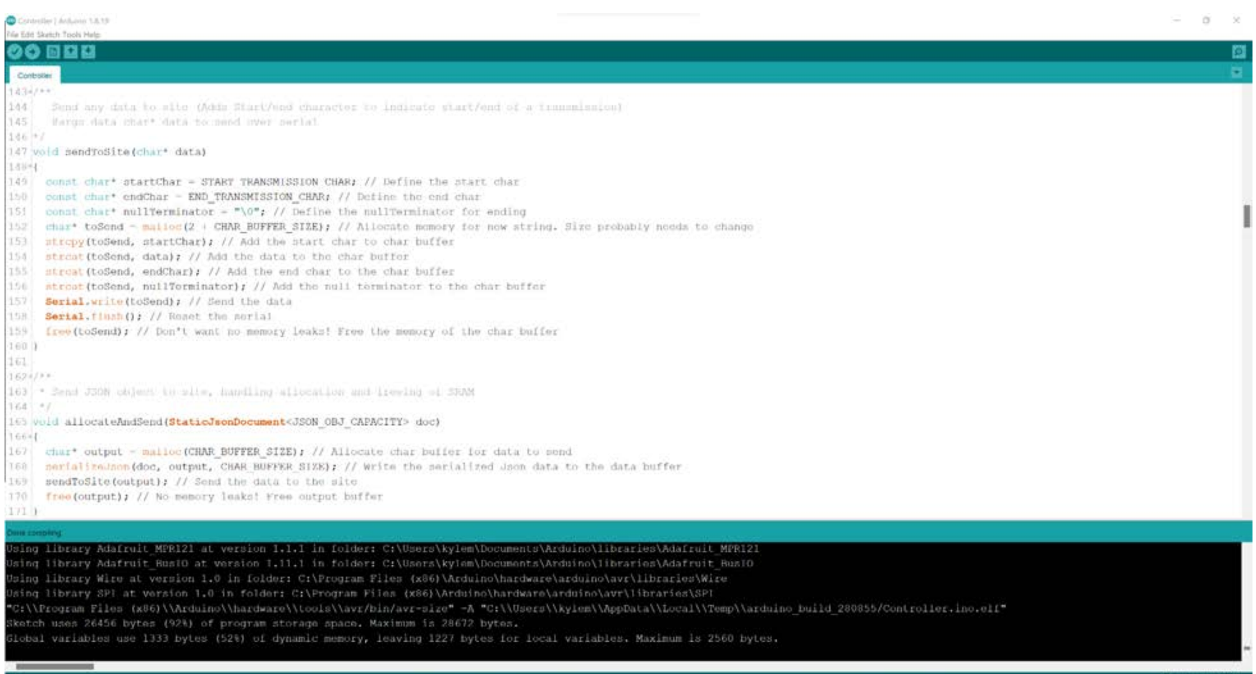

Figure 5.21: Arduino IDE and Code

The Arduino microcontroller's code is always looking for a state change in the main loop function, either from the controller (such as a button press) or for a reception of data from the site. When user input is detected on the controller, the code determines what action has occurred (either a button press, button release, or slider move). After the action has been determined, and the ID of the origin of the action (such as button 1 or slider 2), the software creates a JSON object and gives the necessary key/value pairs that both the controller and website recognises, and sends the JSON to the website.

For sending data to the website, there were several limitations that had to be overcome to send data quickly and properly. The most efficient way to send the JSON data was to use the "Serial.write()" function, passing in an array of the char data type which contained the JSON object. Due to the very limited memory, memory for this char array had to be allocated (using the C malloc() function) and de-allocated (using the C free() command) after every call to the custom send function, with a maximum size of 128 bytes of data that could be sent at once. On top of that, it was quickly discovered that when the website would receive data, it would receive it in bursts that were not necessarily complete. This was due to the way that the serial data is received. The best way to combat this issue was to designate a character that distinguishes the start ("#") and end ("$) of a transmission. These characters allow the website to properly understand the beginning and end of a JSON object. The characters that were chosen to mark the object's beginning and end could be assumed to never appear in any JSON object being sent, however this can be changed easily if one of these characters are ever required.

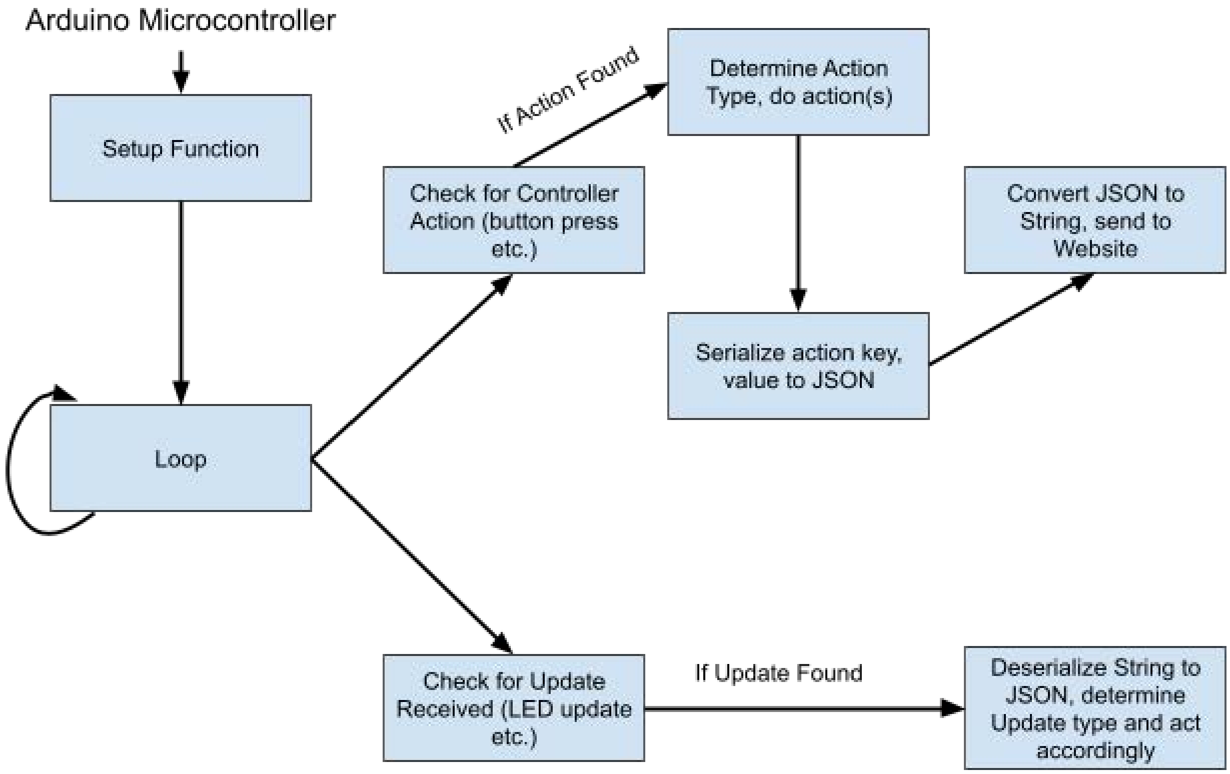

Figure 5.22: State Machine Diagram of Arduino Microcontroller Code

Receiving data from the website was a simpler process than sending it. The only problem associated with this part was the amount of memory the JSON library that was being used takes up. The library "ArduinoJSON" is known for taking up a large amount of memory, thus this problem had to be mitigated. For example, when the JSON data is read from the serial connection, the String data type is converted into an array of the char data type. The use of Strings was minimized in the program because it in particular uses and fragments much more memory than required and would further reduce the already limited usable memory. Once this char array was populated with the JSON data, the "ArduinoJSON" library's "deserialzeJson()" function was utilized to create the JSON object on the Arduino. Lastly, when this "DynamicJsonDocument" is created, the reference is then passed around to the various functions looking through the JSON for any action or update key type in the document.

After the proper hardware was implemented to understand the Slider and Button actions, retrieving user input became the next priority. Using the libraries for the hardware was as simple as waiting for a state change. For example, when a state change was detected on the slider the new value would be recorded and the ID of the slider would be recorded as well. This was placed into a JSON document, and sent to the site. The same thing would occur with the button press actions.

Similarly, when checking for an update sent from the website, the JSON action type will be checked against all the known update types. Depending on which update type, the proper function will handle the update, such as the LEDS update. When an LED update is sent from the website, the Arduino receives the type of LED shape to display, and the Red, Green, Blue (RGB) values. Taking these values, the LED colors are changed using the “FastLED” and “LEDMatrix” libraries which are designed to control the NeoPixels.

6.0 Testing

In this section, we present our procedures and protocols for testing of our prototype iterations. These methods were used for both testing of our paper prototype and final testing of our working prototype.

6.1 Research Methods

In this study, student investigators asked quilters of all experience levels to participate in play testing of various iterations of a soft (fabric) controller to interface with a generative design software. Advertisement to possible participants was done over university email aliases, student organization Slacks and word of mouth using the advertisement seen in Appendix A. For participation in play testing, student investigators informed the participants of the purpose and procedures of the study as well as reviewed the Informed Consent Agreement with the participants answering any questions as they arose. A copy of the Informed Consent Agreement can be found in Appendix B. If the participant consented to participating and signed the Informed Consent Agreement, they continued to participate in play testing of the prototypes.

For play testing of the device prototypes, participants were asked to interact with the controller prototype and use it to interact with the generative quilting design software. A participant who had previously signed the Informed Consent Agreement to be in the study was in a room with the student investigators, all of which were wearing masks. Student investigators began by explaining how to use the prototype and explained any parts of the prototype that at the time had not been implemented yet. Student investigators observed how the participants interacted with the device, as seen in the image below, and asked the participant for verbal feedback and to fill out a feedback survey to influence design modifications for the next iteration of the prototype. The outline of how prototype testing was led can be found in Appendix C.

Figure 6.1: Student Investigators and Faculty Advisor Interacting with Initial Paper Prototype

Observations and feedback were recorded by the student investigators for further reference and for potential inclusion in the final report. The participant was asked if they consented to being photographed while interacting with the prototype and if they consented, student investigators took pictures while the participant interacted with the prototype. At any time, the participant retained the right to withdraw from participating in play testing without any consequence and the right to review any feedback or images of them included in the final report before submission.

7.0 Results and Discussion

In this chapter, we will discuss the results and feedback obtained from our prototype testing. These results were obtained from initial paper prototype testing and final prototype testing through participants interacting and giving feedback on the device following the procedures outlined in the previous chapter.

7.1 Paper Prototype Testing Results and Feedback

For play testing of our initial paper prototype design, we reached out to various email aliases on campus using the email seen in Appendix A. We then explained and collected their signed Informed Consent Agreement, a copy of which can be found in Appendix B, and followed the script in Appendix C for walking the participant through testing the prototype. At the end of the testing we collected user feedback from our participants verbally and through a post testing feedback form.

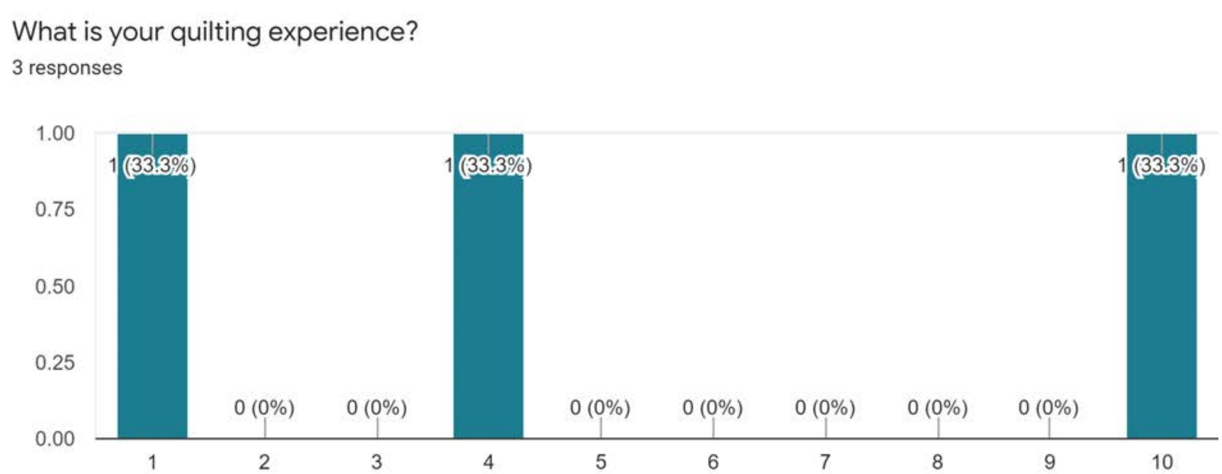

Figure 7.1 Participant Responses on Quilting Experience

Figure 7.1 shows the quilting experience of our participants. As the data shows, we had participants of varying quilting experience which gave us a wide range of input on how our product would work for a beginner, intermediate and advanced quilter.

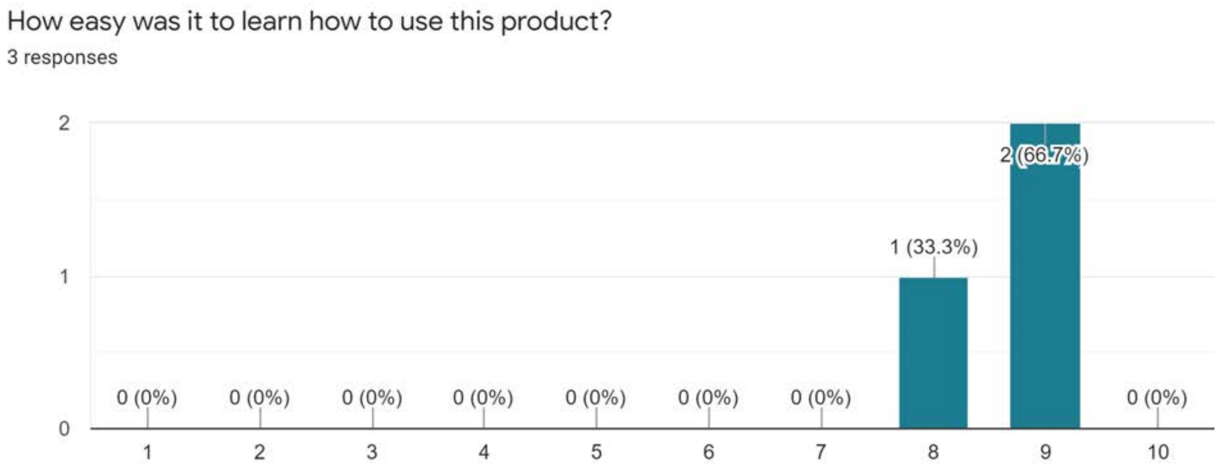

Figure 7.2: Participant Responses of Easiness to Learn

As we can see in Figure 7.2, participants found this device easy to learn how to use. In these tests, we walked participants through an example tutorial of how to use our prototype to design a quilt. Therefore, in future prototypes, it will be helpful to future develop a tutorial to learn how to use this product.

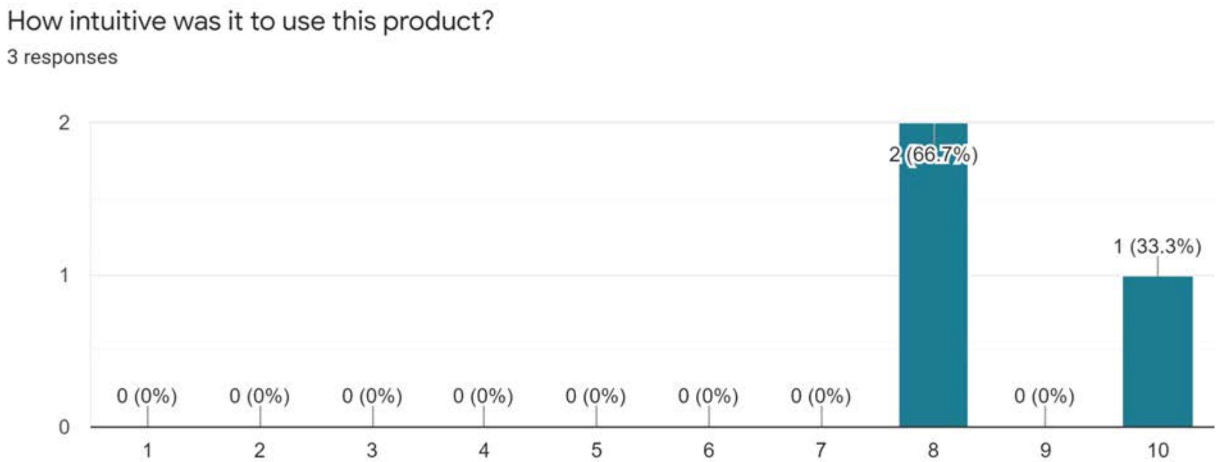

Figure 7.3: Participant Responses on the Intuitiveness of the Product Design

As seen in Figure 7.3 and from participant verbal feedback, we found that the design and layout of the system were very intuitive to use for users. The design is similar to using a keyboard with your computer where your actions on the controller will impact what you see on the computer screen. Since our participants were mainly WPI students, it would be helpful to get more feedback on this from non-technically based people to see if this would be different.

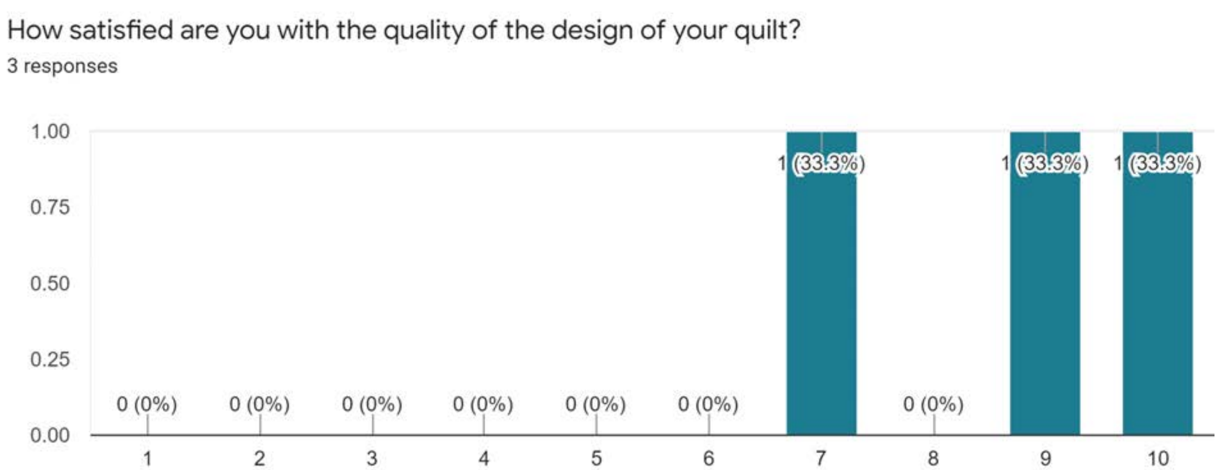

Figure 7.4 Participant Responses on the Quality of their Final Quilt Design

Most participants were very satisfied with the designs that they could create with our device as seen in Figure 7.4. One interesting response we received from our experienced quilter participant was that they would look for much more flexibility in shapes and quilt dimensions in their designs, however, they expressed that more experienced quilters who are looking for that level of flexibility have their own design methods and would probably not be looking to use a device such as ours. From discussing with this participant, it became clear that our best target audience for this device would be beginner to intermediate quilters.

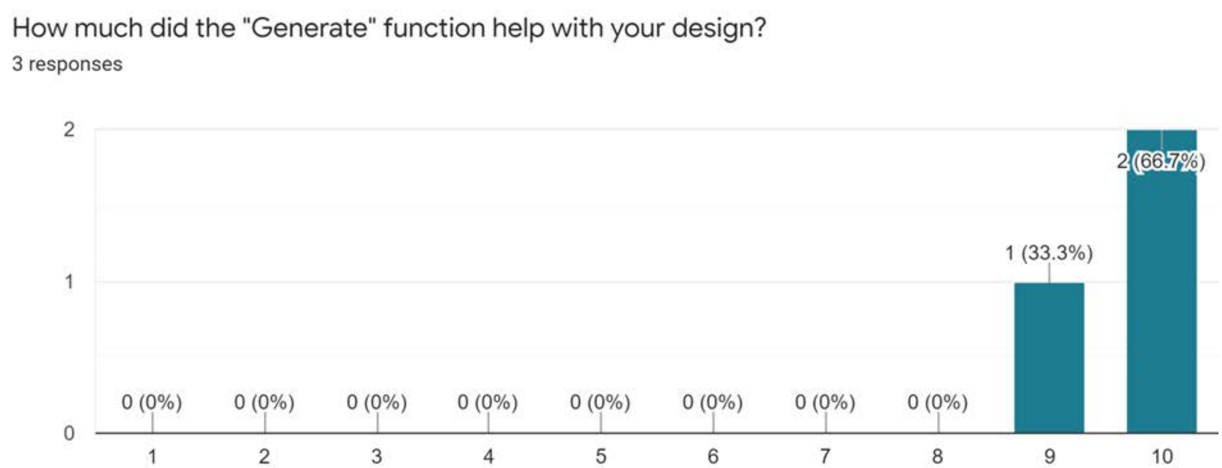

Figure 7.5 Participant Responses on the Generate Function

One significant feature of the existing CodeCrafters program that we wanted to incorporate into our own design was the “generate” function where if the user was stuck, they could randomly generate the quilt. We wanted to add the aspect of being able to save parts of the random generation and re-generate anything that the user didn’t like. As Figure 7.5 shows, participants found that to be a very helpful aspect of this device and it would contribute greatly to their design process.

Some additional verbal feedback that we got from participants included how the sliders were designed. Some participants noted that they had expected three sliders to control RGB but others liked the hue and saturation sliders. All participants noted that they could work with either as long as they were able to obtain the color they were looking for. Another suggestion that was brought up was including a tutorial and pop-up text blocks to tell the user what to do next if they got stuck or confused. A frequent feedback item that was discussed was the possibility of relating the final design to real world materials and dimensions. One participant suggested that with the final design it may be helpful to have the software generate the required yardage that the quilter will need to achieve their design. Finally, a couple of our participants expressed interest in a larger color pallette and shape options to work with. We worked to incorporate some of this feedback in this project but due to the timeline of our project, most of these suggestions would be great to lead future iterations of this project.

7.2 Prototype Testing Results and Feedback

After creating our final device prototype for this project, we had several participants test out our final prototype and provide us with feedback and recommendations for future iterations of this project. In this playtesting, we explained the project and the Informed Consent Agreement for them to sign. At the start and end of the testing session we collected responses from the participants through a standardized survey to gauge their prior quilting experience and their experience with our prototype. In Figure 7.6 below, we have a consenting participant interacting with the prototype.

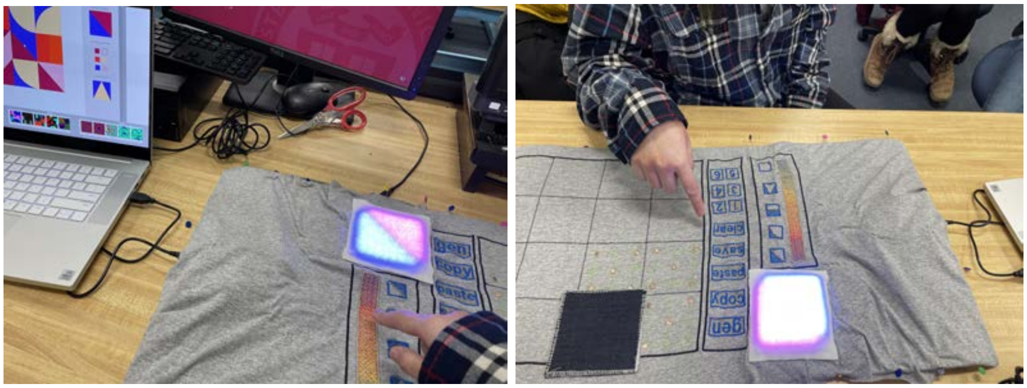

Figure 7.6: Participant Interacting with Prototype

The following images illustrate the responses from our participants about their experience with iterating with the prototype. Figure 7.7 below shows the participant's self reported level of experience with quilting. As evident by this graph, the majority of our participants are novice quilters. In future iterations of this project, it would be valuable to collect feedback from more experienced quilters to get a range of feedback. Unfortunately, due to time constraints we ran out of time to properly advertise to intermediate and experienced quilters. But we were still able to get a couple more experienced quilters in our testing.

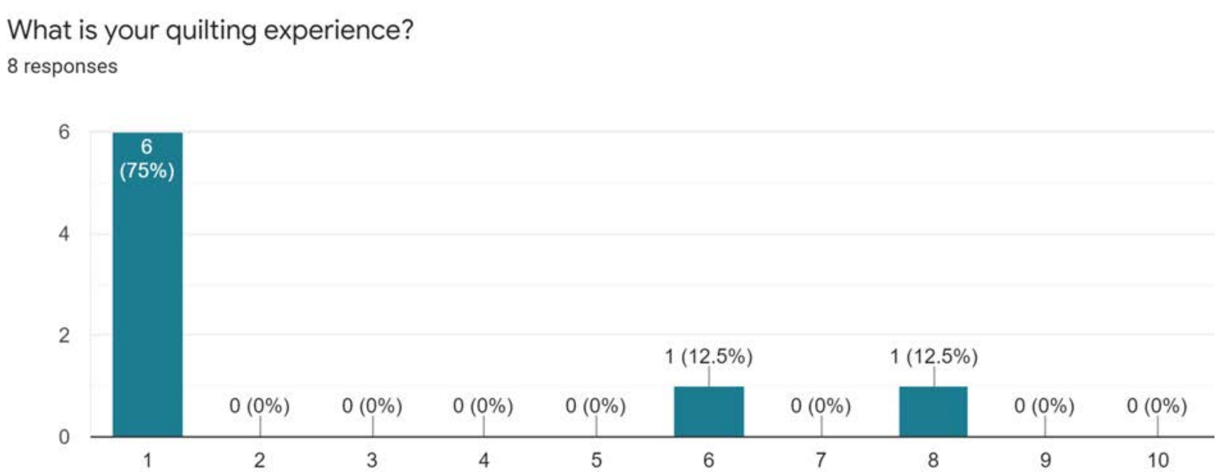

Figure 7.7: Participant Responses on Quilting Experience for Final Prototype

Figure 7.8 and Figure 7.9 below indicate the participants feedback on how easy and intuitive it was to use our prototype. As this feedback shows, the device is fairly straightforward and easy to pick up, however, some participants were initially confused during the explanation of how to use the device. Given this, in future iterations of this project, it would be helpful to have a more in depth tutorial and tips on the web interface to help users quickly learn how to use the device and give hints if the user gets stuck at some point in the design process.

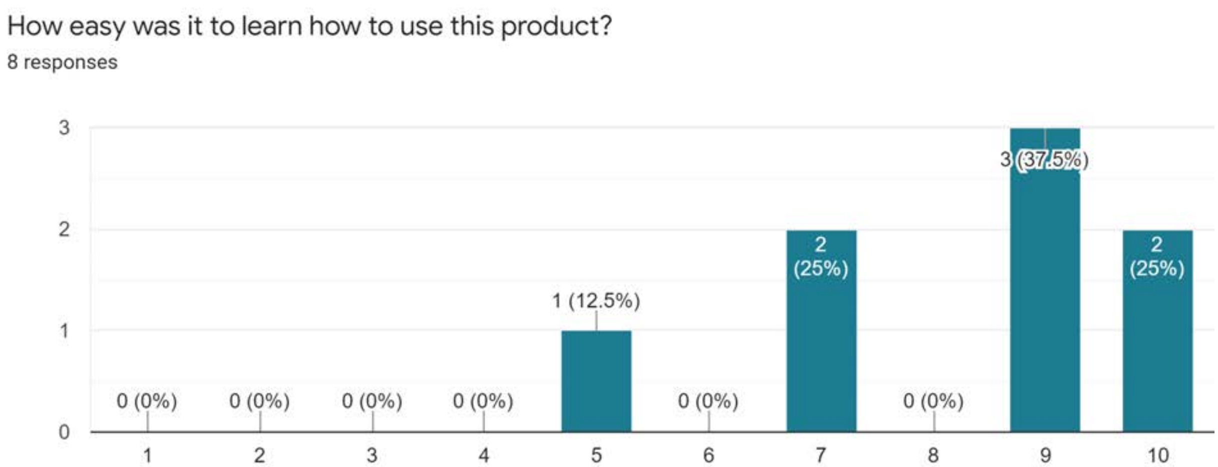

Figure 7.8: Participant Responses on Ease of Learning for Final Prototype

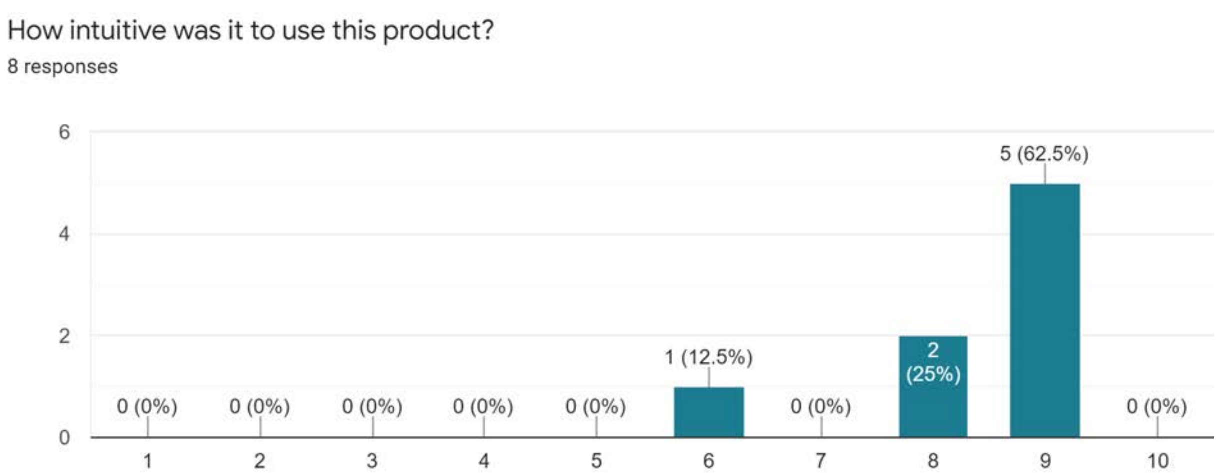

Figure 7.9: Participant Responses on Intuitiveness of Device for Final Prototype

The following image, Figure 7.10, shows the participants' satisfaction with the quality of their design. As this graph shows, the majority of participants were satisfied with their quilt design but there was still room for improvement.

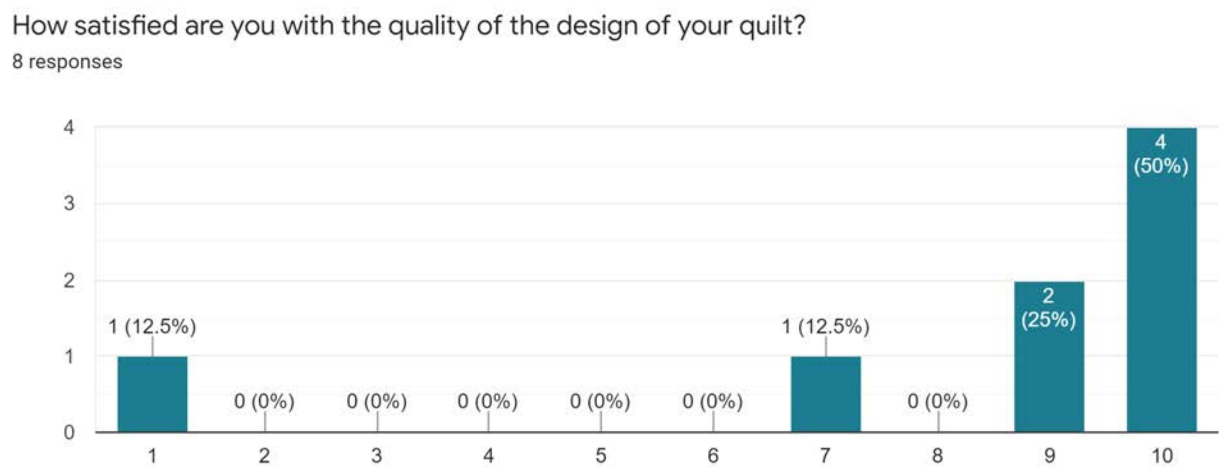

Figure 7.10: Participant Responses on Satisfaction with Quality of Design for Final Prototype

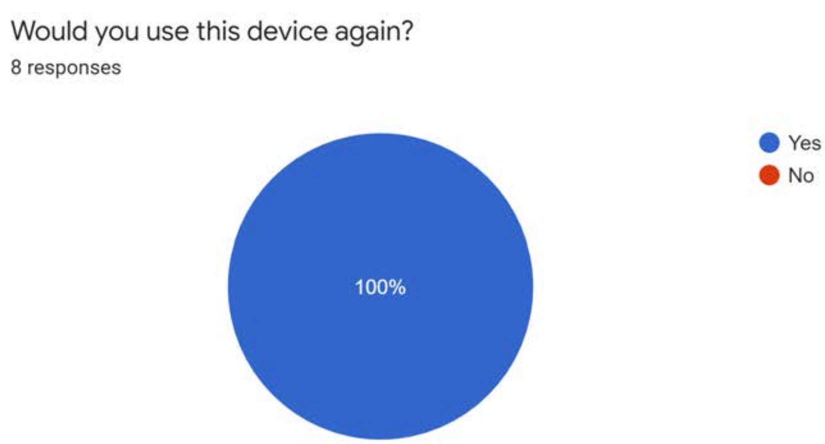

Figure 7.11: Participant Responses on Using the Device Again for Final Prototype

Based on this user testing for the final prototype, we learned that our device is useful and intuitive even for novice quilters. While some functionality that we had originally wanted to include fell out of scope of this project due to time constraints, users were still very impressed with the functionality the device offered and would be interested in using a finalized device of this nature in the future. Based on feedback from the participants, some additional features that they would like to see added in future iterations of this device include better accuracy and range for the color slider, more shape options and more options for block and quilt dimensions.

8.0 Conclusions and Future Work

There was much doubt about the feasibility of this project at its beginning. The main areas of concern were communicating to and from a microcontroller through a web browser, designing a controller interface that would be intuitive and useful, and integrating electronics into fabric, allowing for flexibility. In this chapter, we will explain our accomplishments and difficulties with this project. We also present recommendations for future work on this project.

8.1 Communication and Software

In this section, we will outline the accomplishments, difficulties and recommendations for the communication and software of this project.

8.1.1 Accomplishments

Before starting on this project, it was not known if the specific mode of communication requested was possible, due to the lack of examples on the internet and local professionals lacking relevant experience. However, full communication was eventually achieved with the controller and the website. This feat is a massive accomplishment because it allowed for the proof of concept to be achieved, and the possibility of future work on the project can now be to focus on refining the design, communication, and user experience instead of working on getting the basic communication working.

Furthermore, communications was not only proven to work, the end result shows that the communication is possible without any more hassle on the user's side, including downloading additional software or tools to use the controller. The controller can just be plugged into a USB port, after which a pop-up will be displayed in the browser with the link to the website. Once the user clicks on the popup the tool proceeds to work. The possibilities of this technology will unlock new and exciting hardware designs and software designs through websites and for controllers.

8.1.2 Difficulties

While working on the communications, there were many issues that had to be overcome to achieve the necessary two-way communication required for the controller and website. The first issue was getting the Arduino to be configured correctly for the WebUSB library. Modifications had to be done to the compiler, and this caused many headaches trying to get it to communicate on Windows machines correctly. Furthermore, communication speed currently seems to be slower than anticipated. There is a delay due to the security handshake between the controller and the website. There is most likely a way to further improve this speed, but this was outside of scope of the project. Moreover, memory usage continued to be a problem throughout the development cycle.

The Arduino Leonardo was chosen in part because it was less than $20 a unit (with bulk purchasing options reducing the price or buying non-official microcontrollers). However, it was discovered that the exchange of JSON data proved to use almost all of the available SRAM. This caused many issues when trying to debug because when the program tried to allocate memory that was not available, the Arduino would not offer a warning but instead continued to run the program, with memory that was missing or corrupt. The entire code base was eventually remodeled to reduce memory usage as much as possible using various C techniques to reduce memory usage down to the absolute minimum. More time than was expected was used on the debugging of memory issues, which stalled progress on the controller for a long time.

Furthermore, another hurdle that was presented was understanding the given website code. The code came with some basic documentation in a file, but overall was lacking any in-code comments and did not follow standard coding conventions. This resulted in a significant amount of time spent trying to reverse engineer how aspects of the code worked and how to modify the code to fit its new needs.

8.1.3 Recommendations

There are several recommendations for future projects to work on in the communication and software part of the project. First and foremost, it is strongly recommended to transition to a microcontroller with USB communication that has more memory, such as the Arduino Mega or STM32. Keeping the cost of the Arduino down is still important for this device, but future features will require an Arduino with more memory. Second, reorganizing the CodeCrafters website code could speed up the development process. The code currently functions as expected, but if a future team worked on refactoring the code, using proper codinging standards, Object Oriented Programing (OOP), documenting and commenting as much as possible would greatly advance this project. In addition, cleaning up unused code and styles could assist in further polishing the program up, as many aspects were hidden from the user but still exist and are offered to the user when they first load the page. Third, it is recommended that communication speed become more of a focus as communication between the controller and webpage is unreasonably slow.

In addition, it may be worthwhile to color balance the LED screen, as it currently favors green too strongly and thus does not accurately display other hues. Lastly, further expanding on what can be sent to the controller and what the website can receive can be extremely beneficial to the project and for research. Additional improvements to the patch sensing system, perhaps through the use of different communications protocols would be beneficial, as it would make the device's design more straightforward. On top of this, this WebUSB standard has a lot of potential for experimentation and eventual consumers, and now that the basic communications has been established and proven, further projects can wield this technology for all sorts of applications in the eTextile world.

8.2 Hardware and Embroidery

In this section, we will outline the accomplishments, difficulties and recommendations for the hardware and embroidery of this project.

8.2.1 Accomplishments

In this project, there were many novel eTextile hardware applications put to use that this project student's had never worked with before. It required the significant use of specialized skills, software, and tools by the students in addition to the work of applying these specializations toward this project. Because of this considerable uptaking, it was a tremendous accomplishment to reach even a proof of concept for this device. On top of this, we were able to make quick pivots when unexpected snags were encountered, such as when we discovered issues with integrating the embroidered buttons into the prototype and used the MPR121 breakout board to resolve the problem. In addition to these interdisciplinary approaches and much trial and error, we were able to accomplish a prototype with buttons, a slider, a power supply and an LED matrix working in tandem with a web interface.

Finally, another complex hardware issue that took a significant amount of research and testing was making it possible to obtain the location and orientation of tiles on the design board.

Although we were not able to integrate that with the final prototype and interface with the web browser tool due to time constraints, we were still able to achieve a proof of concept for that functionality of the device. This aspect of the design utilized complex multiplexing and Arduino coding. Given a bit more polish, the feature is capable of being fully integrated with the prototype.

8.2.2 Difficulties

We ran into several difficulties in our design of the hardware and embroidery. The biggest struggle we ran into was the limited time. Given the highly interdisciplinary nature of this project, much of the work on this project was getting the hardware to work individually and then spending a lot of time integrating it with the software. This was the primary reason for the tile position and rotation feature not being fully integrated into the prototype. We were able to get the hardware working for this function on its own but did not have the time to integrate it with the software and web interface.

Additionally, we also ran into the issue of the slider being inaccurate or unable to encompass the full range of color we had hoped for. There were several reasons for this, one being that it was not a priority as we were reaching the end of the project, so it did not get properly debugged. Also, because the original CodeCrafters software has a finite color palette that is based on real life fabric color options, it didn't fit well within a normal sliding scale range.

8.2.3 Recommendations

There are a few places for future work on the hardware side of this project. First of all, as outlined in the previous sections, there is room for improvement and additional functionality in terms of the buttons and slider. With the embroidered buttons and this breakout board, we were able to create 11 functional embroidered buttons and laid the groundwork for an embroidered potentiometer. For future work, an additional MPR121 or clever sensor management is needed to expand this functionality to more input options. There is also the future work of integrating the tile position and rotation functionality with the software and web interface. A future direction for this project could be in cleaning up the CodeCrafters software to better line up with a hardware interface. Finally, in general, there is room to investigate better ways to integrate the hardware components into the prototype in a more flexible way in order to fully satisfy the fabric and flexibility requirements of this project. As eTextile technology continues its rapid development,more and more options for integrating more sophisticated components will make themselves available.

8.3 Final Conclusion

The goal of this project was to develop an eTextile based device to assist quilters in digitizing their designs in a way that was more resource effective and efficient than a traditional design wall, but was still familiar in how users experienced working with it. After successfully integrating our background research, systems engineering concepts, technical knowledge, and research in the fields of computer science and electrical engineering, we were able to create a proof of concept that met these goals but leaves room for future development. A final demonstration of the controller and website can be found at the following link: https://www.youtube.com/watch?v=zIuruJncZHM.

This demo shows our prototype's ability to successfully communicate between the controller, microcontroller, web interface to help users creatively design a quilt patch. With further iterations of this project, this prototype could be fully brought to realization. In short, we were able to create a novel proof of concept for a quilting design quilting design mat to act as a controller for a generative quilting design software to aid quilters in their creative design process by integrating technology without loosing the tactile interface that crafters are already familiar with.

References

About Node.Js. https://nodejs.org/en/about/.

Adafruit Industries. "Linear Softpot (Ribbon Sensor)". Adafruit.Com, 2022, https://www.adafruit.com/product/178?gclid=Cj0KCQiA64GRBhCZARIsAHOLriKiS2cy5IpTwxHyVgwNsVjA6Cnu6R6JGJpv0n1KhZ4N2VgEX71c7ZYaAuZUEALw_wcB

Barker, Sammy. *A Brief History Of The Light Gun On Nintendo*. 2 Nov. 2007, https://www.nintendolife.com/news/2007/11/a_brief_history_of_the_light_gun_on_nintendo.

Beaufort, Francois. *Access USB Devices on the Web*. 25 Feb. 2022, https://web.dev/usb/.

Blanchon, Benoit. *ArduinoJson*. 2021, https://arduinojson.org/.

Brock, David C. “Software as Hardware: Apollo’s Rope Memory.” IEEE Spectrum, IEEE Spectrum, 29 Sept. 2017, spectrum.ieee.org/software-as-hardware-apollos-rope-memory.

Cornish, David. *The History of the Video Game Controller*. 7 Apr. 2021, https://www.shortlist.com/lists/history-of-the-video-game-controller.

Farquhar, Dave. *How Many Atari 2600 Games Are There?* 12 Dec. 2020, https://dfarq.homeip.net/how-many-atari-2600-games-are-there/.

"File:Atari-2600-Joystick.Jpg - Wikimedia Commons". Commons.Wikimedia.Org, 2010, https://commons.wikimedia.org/wiki/File:Atari-2600-Joystick.jpg.

"File:Conductive thread.jpg - Wikimedia Commons". Commons.Wikimedia.Org, 2016, https://commons.wikimedia.org/wiki/File:Conductive_thread.jpg.

"File:Guitar-Hero-Controller.Png - Wikimedia Commons". Commons.Wikimedia.Org, 2008, https://commons.wikimedia.org/wiki/File:Guitar-hero-controller.png.

"File:Leds Built Into Dress.Jpg - Wikimedia Commons". Commons.Wikimedia.Org, 2007, https://commons.wikimedia.org/wiki/File:LEDs_built_into_dress.jpg.

"File:Nintendo-Entertainment-System-NES-Zapper-Gray-L.Jpg - Wikimedia Commons". Commons.Wikimedia.Org, 2016, https://commons.wikimedia.org/wiki/File:Nintendo-Entertainment-System-NES-Zapper-Gray-L.jpg.

"File:Nintendo Labo Construction.Jpeg - Wikimedia Commons". Commons.Wikimedia.Org, 2018, https://commons.wikimedia.org/wiki/File:Nintendo_Labo_construction.jpeg.

"File:"Child's Quilt" - A Jean Ray Laury Design.Jpg - Wikimedia Commons". Commons.Wikimedia.Org, 1970, https://commons.wikimedia.org/wiki/File:%22Child%27s_Quilt%22_-_a_Jean_Ray_Laury_design.jpg.

Fox, Glen. *Guide: What Is Nintendo Labo? 16 Of Your Questions Answered*. 19 Jan. 2018, https://www.nintendolife.com/news/2018/01/guide_what_is_nintendo_labo_16_of_your_questions_answered.

Google Cardboard. https://arvr.google.com/cardboard/.

Hardy, James. *The Beats to Beat: A History of Guitar Hero*. 15 May 2014, https://historycooperative.org/the-beats-to-beat-a-history-of-guitar-hero/.

Hatch Embroidery. 3 Mar. 2022, https://hatchembroidery.com/.

High Conductive Embroidery Threads. 8 Feb. 2022, https://www.madeira.com/embroidery-solutions/embroidery-supplies/industrial-embroidery-threads/technical-threads/high-conductive-threads.

Industries, Adafruit. *FLORA - Wearable Electronic Platform: Arduino-Compatible*. 24 Jan. 2022, https://www.adafruit.com/product/659.

LOOMIA Prototyping Packs & Parts - Quick-Start Pack. 24 Jan. 2022, https://www.adafruit.com/product/4825.

Ivankov, Alex. *Advantages and Disadvantages of Augmented Reality*. 22 Mar. 2019, https://www.profolus.com/topics/advantages-and-disadvantages-of-augmented-reality/.

Kaminsky, Greg. *The Role of Augmented Reality in Military Applications*. 27 Mar. 2019, https://www.ptc.com/en/blogs/ar/the-role-of-augmented-reality-in-military-applications.

Kiourti, Asimina, et al. "Fabrication of Textile Antennas and Circuits With 0.1 Mm Precision." *IEEE Antennas and Wireless Propagation Letters*, vol. 15, 2016, pp. 151–53. *IEEE Xplore*, https://doi.org/10.1109/LAWP.2015.2435257.

Kiourti, Asimina, and John L. Volakis. "Colorful Textile Antennas Integrated into Embroidered Logos." *Journal of Sensor and Actuator Networks*, vol. 4, no. 4, 4, Dec. 2015, pp. 371–77. *www.mdpi.com*, https://doi.org/10.3390/jsan4040371.

"Light-Up Pajamas to Treat Babies With Jaundice." *IEEE Spectrum*, 8 Nov. 2017, https://spectrum.ieee.org/lightup-pajamas-to-treat-babies-with-jaundice.

LilyPad Arduino Main Board | Arduino Documentation. 2 Mar. 2022, https://docs.arduino.cc/retired/boards/lilypad-arduino-main-board.

LOOMIA. *Tale 2 — A History of Smart Fabric*. 26 Apr. 2016, https://medium.com/@LoomiaCo/tale-2-a-history-of-e-textiles-and-conductive-fabrics-dbe9c4a0cb03.

Majaski, Craig. *CES: Now You're Knitting With Power!* 10 Jan. 1987, https://www.nintendotimes.com/1987/01/10/now-youre-knitting-with-power/.

"Make Your Own Design Board | Sewing Room Design, Quilt Design Wall, Quilting Room". Pinterest, 2022, https://www.pinterest.com/pin/342344009160857920/.

"Make A Quilt Design Wall To Preview Your Creations". The Spruce Crafts, 2022, https://www.thesprucecrafts.com/quilt-design-wall-how-to-4145691.

Market, Smart. "Smart Textiles Market Size Global Forecast To 2026 | Marketsandmarkets™". Marketsandmarkets.Com, Sept. 2021, https://www.marketsandmarkets.com/Market-Reports/smart-textiles-market-13764132.html.

Nadalin, Alex. *How Browsers Work*. 29 July 2019, https://www.freecodecamp.org/news/web-application-security-understanding-the-browser-5305ed2f1dac/.

Quandt, Brit M., et al. "POF-Yarn Weaves: Controlling the Light out-Coupling of Wearable Phototherapy Devices." *Biomedical Optics Express*, vol. 8, no. 10, 10, Oct. 2017, pp. 4316–30. *www.osapublishing.org*, https://doi.org/10.1364/BOE.8.004316.

"Quilting Trends Survey Results 2020 - Craft Industry Alliance". Craft Industry Alliance, 2020, https://craftindustryalliance.org/quilting-trends-survey-results-2020/.

"Quilt Market 2021 Outlook, Current And Future Industry Landscape Analysis 2030 – Market Research Blog". Marketresearchblog.Org, 2021, https://marketresearchblog.org/2021/10/quilt-market-2021-outlook-current-and-future-industry-landscape-analysis-2030/.

Red Blob Games. https://www.redblobgames.com/.

"Textile Market Size: Industry Analysis Report, 2021-2028." Textile Market Size | Industry Analysis Report, 2021-2028, Grand View Research, Inc., Mar. 2021, https://www.grandviewresearch.com/industry-analysis/textile-market.

"Textiles: Material-Specific Data | US EPA". US EPA, 2017, https://www.epa.gov/facts-and-figures-about-materials-waste-and-recycling/textiles-material-specific-data.

The Evolution of Guitar Hero Controllers. https://mixdownmag.com.au/news/music/the-evolution-of-guitar-hero-controllers/.

Thrustmaster. *HOTAS Warthog*. https://www.thrustmaster.com/en_US/products/hotas-warthog.

Tyler, Dustin. *Game Programming Languages: The Beginners Guide*. 1 Nov. 2021, https://www.gamedesigning.org/career/programming-languages/.

WebSocket. 12 Dec. 2021, https://javascript.info/websocket.

WebUSB Arduino. https://webusb.github.io/arduino/.

What Is Google Firebase? 26 Nov. 2019, https://blockchainsimplified.com/blog/what-is-google-firebase/.

Wicaksono, Irmandy, and Joseph A. Paradiso. *KnittedKeyboard: Digital Knitting of Electronic Textile Musical Controllers*. p. 4.

Appendices

Appendix A: Advertisement Email for Study Participants

Hello,

We'd like to get feedback on a soft fabric controller we are developing for a student MQP. In this MQP students will be creating a soft (fabric) controller to interface with the live quilting software, CodeCrafters, to assist quilters in their creative design process by allowing users to visualize their ideas digitally before moving to fabric through quick iterations and design generation. We are looking for participants who have moderate to advanced experience with quilting or similar fabric crafts. If you are interested in participating in this study, please fill out this form and we will contact you with further information! Don't hesitate to reach out if you have any questions or concerns!

https://forms.gle/Sy75wvAHTWGefCuq8

Thank you,

Jenna Charron, Kyle Mikolajczyk, Noelle Morgan
eTextiles Interface for Generative Design Software MQP Team

Appendix B: Informed Consent Agreement

Informed Consent Agreement for Participation in a Research Study

Investigators: Jenna Charron, Kyle Mikolajczyk, Noelle Morgan
Contact Information: jcharron@wpi.edu, kjmikolajczyk@wpi.edu, ncmorgan@wpi.edu
Title of Research Study: eTextile Interface for Generative Design Software
Advisors: Gillian Smith, Shamsnaz Bhada

Introduction
You are being asked to participate in play testing of an eTextile Interface. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation.

Purpose of the study: In this MQP, students are developing a quilting device that is designed to assist experienced quilters in their creative design process by allowing users to visualize their ideas digitally before moving to fabric through quick iterations and design generation. It will be created in a way that is familiar to the use of a quilter's "design wall" but is portable, efficient, and more sustainable due to its use of fewer materials than conventional quilting design methods by using components aimed towards wearable devices. The goal of this project is to create an interactive controller device that will pair with a live web quilting software called CodeCrafters. The controller will be designed to send and receive input in an intuitive way for quilters, allowing for physical inputs to result in virtual representations of quilt patterns. Lastly, a secondary goal is to allow the sharing of and collaboration of the users designs to further help with the creative design process.

Procedures to be followed: You will be asked to play test iterations of prototypes of the designed eTextile controller. Testing will involve you interacting with the controller to interface with a live quilting software to test user interaction with the controller. We will inform you of the study's purpose, review and sign this Informed Consent Agreement with you answering any questions or concerns you may have. Once you have consented to participating in this study, we will explain how to use the eTextile controller and live quilting software and observe how you interact with the controller and collect feedback from you. We will ask if you consent to being photographed while interacting with the software. You will have the option to be photographed, photographed with any identifying features excluded from the shot or not be photographed.

Risks to study participants: None

Benefits to research participants and others: Your participation will help inform researchers about how to best design soft controllers for computers.
Record keeping and confidentiality: Records of your participation will be accessible to the study's investigators and advisors. Observations of your participation in testing and photos of your interaction may be included in the project report but your name will remain confidential. Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators, the sponsor or it's designee and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identify you by name. Any publication or presentation of the data will not identify you.

Compensation or treatment in the event of injury: You do not give up any of your legal rights by signing this statement.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact: Investigators (Jenna Charron, Email: jcharron@wpi.edu; Kyle Mikolajczyk, Email: kjmikolajczyk@wpi.edu; Noelle Morgan, Email: ncmorgan@wpi.edu); Advisors (Gillian Smith, Email: gmsmith@wpi.edu; Shamsnaz Bhada, Email: ssvirani@wpi.edu); IRB Manager (Ruth McKeogh, Tel. 508 831- 6699, Email: irb@wpi.edu) and the Human Protection Administrator (Gabriel Johnson, Tel. 508-831-4989, Email: gjohnson@wpi.edu).

Your participation in this research is voluntary. Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit.

By signing below, you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

____________________________ Date: ____________________

Study Participant Signature

____

Study Participant Name (Please print)

_____________________________________ Date: ____________________

Appendix C: Paper Prototype Testing Script

Introductions/Consent Form

1. Introductions, thanks for participating!
    a. Explain what each of us will be doing during the testing (who will be recording, who will be taking notes, who will be leading, etc.)
2. Go over consent form with participant (bring printed form)
    a. Explain "purpose of study" section primarily
    b. Answer any questions they have and have them sign
3. Have participant fill out consent to photography and video section of Follow-up Form

Explaining Prototype

1. This board acts as a controller for this quilting design wall software. The purpose of this prototype is to assist in creating a design wall digitally. What you do on the controller will be reflected in the software on your computer in the final device. For today, we have a paper controller and mock up software screen to get a better idea of how you would interact with the final device. We will narrate for you what your actions on the prototype will produce on screen and give you guidance throughout the process.
2. On this controller, you have many "patches" (sticky notes currently) that emulate your fabric patches when creating a block. These patches can be moved to the "Designer Block" to add shapes and colors to these patches. There are several buttons and sliders that are used in the design process that we will explain through the tutorial and we have a switch up top for when you want to switch between working on a block or a whole quilt.

Designing a Block

1. Clear a square of patches starting from the top left out to the dimension you want for your block. (to match our mock-up UI, clear 9 patches)
    a. This determines the size of your Block and the LEDs on the edges will light up to show you your "active" dimensions
2. Your "Working Palette" will be the colors that you will be working with to design your Block
    a. To change your working palette, you can either drag a saved palette over to the working palette area to use that existing palette
    b. Or you can press and hold the corresponding "Palette/Block" button that you want to change, then once it is highlighted on the screen, you can use the "Colorpicker and Greyscale" sliders to cycle through your color options until you reach the color you want
    c. To change another color, press and hold another button and do the same thing
    d. Once you have the "Working Palette" you like, press "Save"
    e. You can change these colors anytime during the Block design process by pressing and holding a "Palette/Block" button, however, changing a "Working Palette" color will change that color on the blocks you have already assigned this color to.
3. Designing Patches

    a. To design a patch, place one of your patches (sticky note) on the Designer Block with the notch in the top left corner
    b. This patch will now be reflected on screen on the top right and your design choices for this patch will be roughly reflected with LEDs on the controller
    c. First select your "Patch Shape"
    d. Then on screen, you will see one half of the Patch highlighted. The area highlighted is the half of the Patch that you are selecting a color for.
    e. Using the "Palette/Block" buttons, chose the corresponding button to the color you want on the highlighted part of the Patch
    f. To choose the other part of the Patches color, hit the same shape button again and the other area of that Patch will be highlighted then select the Palette Color that you want on that half
    g. If you want to edit the color on the other half again, hit the shape button again or if you want to chose a different shape, click a different shape button
4. When you are happy with a designed Patch, place it on your "Design Wall" in any active area in any orientation - this placement will be reflected on screen
5. At any point you can copy/paste any Patch you have by putting the Patch you'd like to copy on the Designer Block and clicking the "Copy" button
    a. Then this Patch will be seen in your "Clipboard" on screen
    b. You can put that Patch back and take a new Patch and place that new Patch on the Designer Block and click "Paste" to paste the Patch design in your Clipboard to that new Patch and place that new Patch on the board
6. At any point, you can clear your Design Wall by pressing Clear and removing all Patches
7. At any point, when you have a Patch on the Designer Block, you can click "Generate" and that will generate a random patch design
8. If your Designer Block is empty, you can click "Generate" and that will randomly generate your Block using your Working Palette
    a. Any sections covered in a Patch, will be considered "locked" and will not randomly generate
    b. If you like part of the generated block, you can place a Patch over that square to "lock" that random Patch in
9. When you are finished with your Block design, press "Save"
    a. This will save your block to the community tab and save it to use on your quilt design
10. Now you can either start a new block by clearing your Design Wall or you can toggle the switch at the top right of your controller to Quilt to design your quilt

Designing a Quilt

1. After you have used the switch to go to designing a quilt, your screen will show you the "Quilt Design" UI

2. To start, remove the Sticky notes to the dimension you want your quilt to be, this dimension will be reflected on screen and with LEDs on the controller
3. On the left of the screen, you will see “My Blocks” which is all your saved Blocks to pull from for this quilt
4. You will be working with your “Working Blocks” to design your quilt, so drag and drop on your laptop, the Blocks from My Block or Community Blocks that you want to use in this quilt
5. To place your blocks, start by putting on of your Blocks (Sticky notes) into the Designer Block, then press the respective “Palette/Block” button corresponding to the Working Block you want and place that block on your design wall in any orientation
6. Or you can use the Generate button to randomly generate your quilt using the blocks in your Working Blocks section
    a. Same “locking-in” rules apply as from before
7. Once you finish your quilt, press “Save” and you’ve successfully designed a Quilt!

Follow-up

1. Ask participant for any feedback and have them finish their Follow-up form
2. Send them a copy of their signed Consent form for record and contact information
3. If we use any of their pictures or direct quotes we will contact them to review those sections before final publication of our report
4. Ask if we can contact them for future prototype iteration testing in C-term

Appendix D: Code

Use this link to access the Arduino code for this project:
https://github.com/kylemikableh/eTextilesArduino

Additionally supplementary code will be attached on eProjects as supplementary material.