

TEXELS: A PROGRAMMABLE TEXTILE INTERFACE FOR REPLICATING  
TEXTURES

by

MAYA EUSEBIO

A thesis submitted in partial fulfilment of the requirements

for the Honors Undergraduate Thesis program

in the College of Engineering and Computer Science

and in Burnett Honors College

at the University of Central Florida

Orlando, Florida

Spring 2026

Thesis Chair: Joseph LaViola II

© 2026 Maya Eusebio

## **ABSTRACT**

Self-moving fabric interfaces have massive potential for applications in fields ranging from art to haptic feedback to deployable space structures. However, current systems of implementation face the impracticalities of bulkiness, burnout, and energy consumption on top of limiting designs that can only contract uniformly or create one pre-programmed shape. For this technology to bring the change that it promises, we must break the barriers of usability and sustainability to make it a practical choice. This thesis aims to develop a scalable model of fabric that designers, programmers, and anyone else can acquire or create with accessible materials, integrate into design flows with traditional machinery, and reprogram into any shape they need as many times as the textile can be reused. To do this, we develop a universal hinge system based on the concept of pixels that can be used with input data to replicate a myriad of different textures via smocking. Increasingly complex prototypes are theorized, implemented, and discussed.

To the student reading this. If I could do it, so could you.

## ACKNOWLEDGEMENTS

I would like to thank the University of Central Florida's College of Engineering and Computer Science, Burnett Honors College, Office of Undergraduate Research, and Research and Mentoring Program for providing me with funds to support this research. The encouragement and enthusiasm I have received for my work from the directors of these programs has consistently reignited my passion.

In addition, I am grateful for the space I received at the Interactive Systems and User Experiences lab to learn about research and explore my own curiosities as far as I dared to take them.

I am lucky to have the unconditional support of my parents. Thank you for showing me that I can climb as high as I want to, in any direction, and with as many curves in my path as I choose. Thank you for everything you do to keep a net under me as I do so.

I'd like to acknowledge my friends who kept me grounded through this process. Those who gave me life to look forward to in person, those who gave me faith in myself from time zones away, and those who continuously inspire me to dream bigger and go farther. I want to be like you all when I grow up.

Finally, thank you to my chair Dr. Joseph LaViola II, my committee members Dr. Joshua Kaufman, Dr. Raj Vaidyanathan, and Dr. Canan Dagdeviren, and the inspiring professors who have modeled their versions of success to me and supported me in finding my own.

This work was conducted in the spirit of the Slow Science Manifesto, advocating for collaborative and sustainable research ([slow-science.com](http://slow-science.com)).

# TABLE OF CONTENTS

<b>LIST OF FIGURES</b> .....	<b>vii</b>
<b>LIST OF MEDIA, SYMBOLS, ABBREVIATIONS, NOMENCLATURE, OR ACRONYMS, ETC.</b> .....	<b>viii</b>
<b>INTRODUCTION</b> .....	<b>1</b>
Motivation.....	1
Problems .....	1
Contributions.....	2
Reader’s Guide .....	3
<b>RELATED WORK</b> .....	<b>5</b>
Shape Language .....	5
Actuation Methods .....	10
Materials .....	11
Summary of Choices and Advantages .....	14
<b>DESIGN RATIONALE</b> .....	<b>15</b>
Overview .....	15
Subsystems.....	16
<b>PROTOTYPING</b> .....	<b>29</b>
Preliminary Prototype .....	29
Final Design .....	32
Implementation .....	34
<b>DISCUSSION</b> .....	<b>37</b>
Strengths of Current Prototype .....	37
Limitations and Recommendations for Further Development .....	37
Future Evaluation and Applications.....	39
<b>CONCLUSION</b> .....	<b>41</b>
<b>APPENDIX A:</b> .....	<b>42</b>
IRB APPROVAL LETTER .....	42
<b>APPENDIX B:</b> .....	<b>44</b>
ARDUINO CODE .....	44
<b>APPENDIX C:</b> .....	<b>49</b>
USER INTERFACE CODE .....	49
<b>APPENDIX D:</b> .....	<b>56</b>
NORTH AMERICAN SWATCH SAMPLES .....	56
<b>APPENDIX E:</b> .....	<b>58</b>
VIDEOS AND ADDITIONAL MEDIA OF PROTOTYPES.....	58
<b>REFERENCES</b> .....	<b>60</b>

## LIST OF FIGURES

Figure 1 - Miura Ori tessellation folded from paper. At rest (left), with one cell flattened (middle), and with one row flattened (right). .....	7
Figure 2 - Waterbomb tessellation folded from paper. At rest (left), pulled into a cylinder (middle), and pulled into a hemisphere (right). .....	8
Figure 3 – All attempted Nitinol shapes (at rest above, actuated below). Bending straight upwards with an eyelet for sewing (A), bending at a 90 degree angle with flat ends (B), bending into a downward arch (C), and coiling into a spring (D). .....	19
Figure 4– Single wire actuation circuit modeled on Simulink.....	20
Figure 5– Four individual wires arranged into a texel unit (A), and an example circuit for a two-dimensional array of texels (B) modeled in Simulink.....	21
Figure 6– Control unit schematic. An Arduino ESP32 Nano connected to five shift registers. The first shift register serves as a row driver while the remaining four are chained together as column drivers.....	22
Figure 7 – Mockup of pattern-making software user interface. ....	26
Figure 8– Mockup of library selector software user interface.....	27
Figure 9– Schematic of initial prototype (A) with closeup of techniques used to connect each SMA wire (B). .....	29
Figure 10– Testing of preliminary prototype. At rest (left) and 5 seconds after selecting actuation (right). .....	30
Figure 11– Schematic of the final layered design.....	32
Figure 12– Closeup of sewn connections of one Nitinol wire.....	33
Figure 13– Testing of the layered iteration. At rest (left), engaging with interface to actuate (middle), and 3 seconds after selection (right).....	34
Figure 14– Closeup of sewn connections implemented physically. ....	34
Figure 15 - Wires 1-4 activated for wave pattern.....	35
Figure 16 - Wires 2, 5, 6 activated for brick pattern.....	35

**LIST OF MEDIA, SYMBOLS, ABBREVIATIONS, NOMENCLATURE, OR  
ACRONYMS, ETC.**

<b>Term or Symbol</b>	<b>Meaning</b>
2D	Two dimensional
3D	Three dimensional
NURBS	Non-Uniform Rational B-Splines
SMA	Shape Memory Alloy
SMP	Shape Memory Polymer
LCD	Liquid Crystal Display
PCB	Printed Circuit Board

# INTRODUCTION

## Motivation

Programmable textiles are fabrics which can be made to perform a certain function or arrange themselves into a certain physical appearance. In this thesis specifically, we discuss the subgenre of programmable textiles that can configure themselves into a predetermined shape to fulfill their purpose. There are many cases where a fabric like this could come into use. For example, art installations and environmental set designs in the world of User Experience could benefit from being made of materials that actively respond to the people observing them [9]. Such fabrics could also replicate real-world textures or provide versatile tactile cues for more accurate haptic feedback systems [6]. Fabrics that self-assemble could become deployable structures that increase efficiency in space exploration or disaster response [25]. Additionally, the fashion industry could see big changes with this technology on the horizon. Not only would there be a creative boom in possibilities for high-fashion exhibitions that move or feel more alive, but at the consumer level it could become more common to see custom-fit pieces and assistive clothing that exist in a symbiotic relationship to the wearer [2]. So long as we can make it a practical option, the potential impact of programmable shape-changing textiles is limited only to the imaginations of the designers and scientists who wield it.

## Problems

Despite this potential, current implementations of programmable textiles face many challenges to overcome before they can go on to practical consumer use. Some textiles have only one actuated state and must be manually reset, while others only expand and contract. All of

these solutions are greatly limited in the number of textures that can be replicated by a single fabric. Materials capable of actuating the textile have also historically been too rigid to integrate into textile surfaces comfortably [7]. Some materials take longer to activate than others, which can cause inconvenience. Production can become costly with novel materials or fabrication methods. The product can be stiff and inefficient if too thick, fragile if too thin, and uncomfortable for users on either extreme. Materials that burn out quickly may render the textile obsolete in a short number of uses. Using electricity as an activation energy necessitates a power supply. These challenges could all halt production if not addressed appropriately.

Even if a textile is successfully produced, some issues can still prevent its use in the real world. For example, some systems may rely on complex digital design tools that create a barrier to entry for users. Fabrics that are incompatible with traditional machinery and practices such as sewing and washing machines are less likely to be used. Some projects describe systems for users to create their own programmable textiles, necessitating machinery and materials that they otherwise would not have, rather than disseminating the textile itself as a raw material. Finally, products made from unsustainable materials will become less feasible to create and consume in the future.

### **Contributions**

Although no single project would be able to address all of these concerns at once, we believe ourselves to be the first to tackle several of these issues. We propose a shape-changing textile interface that is:

- Capable of multiple actuated states, and thus a wide range of programmable shapes and textures
- Reprogrammable into different states multiple times.
- Intuitive for users, gathering necessary data from simple selection inputs and usable without explicit training in engineering or fashion
- Accessible to create with common components and techniques

Contingent on the success of several increasingly complex prototypes, the proposed textile would be able to advance the current state of shape-changing textiles into something much more feasible for consumer use, so that designers in any field can program matter to their needs.

### **Reader's Guide**

This thesis is composed of six chapters and five appendices.

Chapter two: RELATED WORKS reviews the literature on active materials, specifically the shape language, actuation methods, and materials that comprise them. Related works are discussed.

Chapter three: DESIGN RATIONALE gives an overview of the proposed system and details the choices made in creating each of its sub-units.

Chapter four: PROTOTYPING shows the schematics and physical implementation processes for two prototypes of the full system.

Chapter five: DISCUSSION sheds light on the strengths and limitations of the final prototype design, recommends improvements for continued development, and introduces future evaluation techniques and applications.

Chapter six: CONCLUSION summarizes the work done in this thesis.

APPENDIX A contains the IRB approval letter for a future user study of the device.

APPENDIX B contains the code used to program the control unit of the device.

APPENDIX C contains the code used to generate the software interface for the device.

APPENDIX D gives sample patterns of the smoking technique employed by the device.

APPENDIX E contains a link to videos where movement can be observed during testing.

## **RELATED WORK**

A review of the literature to present day was crucial to positioning this work at a novel perspective for the possibilities of shape-change fabrics. There are three components to the creation of a programmable textile, and many ways of mixing and matching approaches to those components to realize a full system. First, there is the shape language of the system. This underlying geometry determines the type of shapes that the textile can replicate, how accurately it will do so, and how pieces can scale together into larger forms. Next, there is the method of actuation. In general terms, this is how energy will be harvested and used to make the garment change shape. Then, materials must be chosen that can carry out the actuation method while maintaining a standard of comfort, compatibility with traditional textile processes, durability, etc.

### **Shape Language**

The ability to change shape is a promising aspect of both textile-based and non-textile-based programmable matter. One tactic for changing shape is to have subunit modules that are capable of physically rearranging themselves into 3D structures. The term ‘claytronics’ was coined to describe this strategy [26]. When it comes to fabrics, however, this method is insufficient as the textile must begin as a sheet of woven fibers and maintain that structure through the process. Thus, we rely instead on ways to approximate 3D shapes and textures through the deformation of a 2D surface.

There are two broad categories of geometry explored in the literature to render 3D shapes out of 2D surfaces: Polygonal modeling and Non-Uniform Rational B-Splines (NURBS) modeling.

### *Polygonal*

Polygonal modeling represents complex shapes by combining subunits of the same basic 2D shape (often triangles) at different angles and sizes. The resultant shapes are less true-to-life, as it takes infinitely many two-dimensional polygons to mimic a three dimensionally curved surface. Actuation involves treating subunit edges as “crease lines” with a limited number of activation states. These states can be binary (“folded” and “unfolded”) or they can be set between “mountain,” “flat,” and “valley.” Simple options streamline the process of communicating patterns to the garment, though the final shape may carry a level of inaccuracy.

The best approach to creating 3D polygonal shapes from 2D planes via surface deformation is by referencing the traditional art of origami. To date, algorithms have been defined with the capability of defining an origami pattern for any 3D shape [26]. To translate such an algorithm to fabric, however, there would need to be a way of transforming a theoretically infinite amount of weft, warp, and diagonal paths between woven fibers into crease lines and actuating them selectively. Additionally, when algorithms like these are implemented with physical materials having some level of thickness, even though they might be optimized to create as few layers as possible, resulting structures could become impractically bulky. For example, Benbernou et. al. take a box-pleated universal hinge pattern that they proposed in 2009 [33] and deduce that in the worst case, a sheet of area  $\Theta(N^2)$  can have up to  $\Theta(N^2)$  layers folding together at a given point [2]. They propose instead a strip-based solution with unfolded units that are only long in one direction. This approach reduces surface area waste, but results in shapes

with lower structural stiffness. Applied to a textile context as opposed to rigid robotic frames, resulting structures would be unlikely to hold their shape.

An origami-based approach that also seems viable is a multi-stable pattern that can in turn be contorted into several shapes and scaled through repetition of the pattern. Two origami patterns in the literature are assessed for their fitness to these requirements.

### **Miura-Ori Tessellation**



Figure 1 - Miura Ori tessellation folded from paper. At rest (left), with one cell flattened (middle), and with one row flattened (right).

The Miura-Ori tessellation has been used in many cases for its auxetic properties [22, 24, 27]. Once fully folded, it can exhibit something close to multi-stability in that a row of mountain folds can be flattened as a unit with the rest of the structure remaining intact. Triangular cells can also be flattened without much effect on their surrounding columns or the rest of their row of folds. Aside from this, however, the height of these deformations can only be binary between flat and the original fold's height. Therefore, rather than being able to make a variety of shapes, the Miura-Ori fold is better suited to creating a surface of variable roughness on a small scale.

## Waterbomb Tessellation



Figure 2 - Waterbomb tessellation folded from paper. At rest (left), pulled into a cylinder (middle), and pulled into a hemisphere (right).

Waterbomb tessellations can more easily replicate NURBS-like shapes. Their fully folded configuration results in a flat surface, while pulling at the edges results in a hemispherical curve. This curve, however, is not bidirectional. The tessellation can only be fully convex or fully concave. Additionally, making cuts to this tessellation would be unintuitive to users, as one must cut along specific fold lines for the cut to appear straight when folded without compromising the pattern.

## *NURBS*

NURBS curves are also created from polygonal subunits, but with a different activation strategy. In the case of NURBS, edges are actuated via curling movements [12, 17] rather than folding. These curls result in smooth surfaces that can more accurately depict organic shapes.

## Incremental Actuation: Edges

Theoretically, there is an infinite number of actuated states for each edge ranging from completely flat to completely bent at any point along the edge. For example, SMAAD and

NURBSforms use square or triangular subunits of fabric with shape memory alloy wire that can be sent varying levels of voltage to communicate bilaterally with modeling software. The size of these subunits, however, renders them impractical to render complex shapes or textures. The computations to render these shapes also become increasingly complex as the amount of actuation states that can be sent to each wire increases. One must also consider that scaling these designs to combine multiple polygons must include a control unit for each individual polygon or otherwise effectively send different levels of actuation to two edges that may be connected in the form of a continuous line.

### **Binary Actuation: Smocking**

One may be able to reap the benefits of both organic NURBS shapes and binary actuation options through traditional sewing techniques. Smocking is a method where two points on a fabric are drawn and affixed together by knotted thread, causing the space between to bunch into a continuous curve [28, 39]. Many shapes and patterns can be embedded into the surface of a textile through strategic connection points. Although freeform smocking allows for the greatest range of resulting textures, replication of this technique suffers from similar complications to origami in that any two points on the fabric plane must have a way of connecting. North American smocking is a subset of this technique which uses a standardized grid to determine which points can be connected [39]. Although a fixed grid size limits the potential height of smocked curves, there is still a great variety in texture options that can be replicated by connecting solely points established on the grid. For examples of some established North American smocking patterns and their sewn counterparts, see Appendix D.

## **Actuation Methods**

Depending on a textile's intended uses and material composition, there are many suitable options for actuating shape change. One important distinction for this project is that actuation strategies can be categorized as reversible or irreversible. To make our fabric reusable, we will focus solely on methods that can be reversed.

Actuation methods can also be categorized broadly by the activation energy that is used to carry them out. Among these are heat, light, fluid pressure, simple kinetic force, hydration, and sound [2]. To replicate 3D textures, the fabric must be actuated in very precise locations. This fact places light, hydration, and sound outside of our scope of interest as they are difficult to control and could also result in an unpleasant experience for the user if incorporated into a garment.

### ***Kinetic Force***

Simple kinetic force can be used as an activation energy without requiring the use of novel materials. Traditional textiles can be made to change shape through the introduction of a force or removal of a load-bearing element, affecting the entirety of the connected textile sheet. This method is advantageous as it is the most likely to be compatible with existing textile technology and requires little to no additional accessories (such as pumps or power supplies). Designing multi-stability into a system like this can be done but often allows for multiple stopping points along a single transformation from one shape to another (e.g., a drawstring skirt with adjustable length) as opposed to multiple differing shapes.

## *Heat*

Heat activation energy can be deployed in a precise manner via electricity or heat guns. Tessmer and Tibbits developed a knit mask that would contract wherever heat was applied through a heat gun on a programmable robotic arm [33]. This process, however, relies on a large accessory that could be hard to access. The changes to the textile were also irreversible. Projects like Chromorphous [4] send an electric current through conductive wires woven into a textile to generate the heat needed for the material to change. This method is reversible and more consumer-friendly but still necessitates a power supply and control unit. One must also carefully mitigate the current's potential to burn, short, shock, or otherwise cause harm to the garment or its wearer.

## *Fluid Pressure*

Fluid pressure is used in projects like OmniFiber [11] to facilitate shape change by causing threads to expand and contract in diameter. An external pump is necessary in this case to circulate the energy of actuation, and this can be a bulkier accessory to require than an electrical power supply.

## **Materials**

Many different types of materials have been explored for their potential applications in programmable textiles. Materials can either be categorized as active or inactive. Additionally, projects that must use a combination of active and inactive materials may combine them on a textile scale or at the fiber level.

### *Inactive Materials*

Inactive materials encompass most traditional textiles, which do not respond to activation energies outside of kinetic force or at times fluid pressure. In creating a programmable textile, these materials must be conjoined with an active material in some way to make the piece responsive to a different activation energy. The most important factor to consider when choosing an inactive material is whether it is elastic, inelastic, or auxetic, such that it can move properly alongside its active counterpart(s).

### *Active Materials*

Active materials can respond in some way to activation inputs such as heat and light. For the purposes of this project, we will consider only materials whose response to activation is reversible movement, and which can be activated by heat.

FibeRobo is a liquid crystal elastomer fiber that can contract when exposed to heat, and naturally relaxes once this heat is removed [5]. Fibers that expand and contract can be woven into inactive textiles in many ways to produce different shapes when they are activated. Once they are woven, though, these textiles are constrained to one activated shape for their lifetime. ModiFiber is another string actuator made with twisted-then-coiled nylon and silicone. It is similarly limited in that it can only contract or twist when exposed to heat [10]. It also actuates much more slowly than other options. Shape Memory Alloy (SMA) and Shape Memory Polymer (SMP) are more common choices for reversibly heat-active materials. They hold an advantage in that they can be trained into a variety of actuated states, rather than just expanding and contracting. SMA has the well-documented disadvantages of being quick to burn out, costly, and

complicated to train. It can also be too rigid for a comfortable textile experience. SMP, while softer, is typically not electrically conductive and thus must be activated by an external accessory applying heat. For the purposes of this thesis, the benefits of electrical activation that can be integrated into the device itself for a more user-friendly experience outweigh the inconveniences of training and cost. Therefore, SMA is preferred.

### ***Textile Scale Integration***

The simplest method to combine multiple materials is to use techniques already common in traditional fiber manipulation. For example, when materials can be acquired in or formed into the shape of long fibers, weaving is a common way to accrue them into a full textile sheet. One could also consider knitting or crocheting to create textile sheets, as well as sewing and embroidery to bring materials together in a nonuniform manner. Although less common in traditional garment-making, 3D printing can also be a quick and effective way to integrate textile and nontextile materials, or lay nontextile materials into a sheet [2, 3, 13]. These techniques are the most easily accessible and thus lend themselves readily to rapid prototyping and early development.

### ***Fiber Scale Integration***

Fiber extrusion and drawing are processes by which a single fiber can be designed as a thick cylindrical core and pushed or pulled out into a thin thread with the same internal composition. The Chromorphous project, for example, uses fiber extrusion to create its threads with a conductive core and polymer outer layer, before combining those fibers into one uniform sheet of woven textile [4]. While combining materials at this scale leads to a very seamless

finished product, it requires the use of expensive machinery that often also needs to be modified to work with nontraditional materials. Therefore, it serves as an avenue worth exploring only after the textile's functionality has been verified by a prototype.

### **Summary of Choices and Advantages**

Overall, the advantages of the choices made for this thesis are three-fold. Firstly, a variable smocking technique has not yet been explored as a shape language for an active textile. This design allows for smooth curves and different textures while keeping the systemic simplicity of binary actuation modes. Second, the use of shape memory alloy as an active material allows for a self-contained system that does not rely on external tools for actuation. Finally, the use of common sewing and embroidery techniques to integrate materials allows for rapid prototyping and validation of the whole system before involving complex or expensive machinery.

# DESIGN RATIONALE

## Overview

Similarly to how LCD screens use individually addressable pixels to produce multicolor images, our fabric will have individually addressable grid squares that can be smocked in different configurations to create a texture, referred to from this point on as “texels”. Where pixels have red, green, and blue subpixels, our texels will be subdivided into the lines which connect each corner of the grid square. North American smocking patterns can then be reflected onto the fabric grid. Lines designated as smocked in a given pattern will be sent an electric current to smock them.

Individual texels will be addressed with drivers in the same way that pixels are addressed on screens. A row driver will first open the gates for each row sequentially. Then, a column driver will send the appropriate voltage to each texel in that row. Sending energy to a texel will activate it, causing the embedded active material to contract and draw together the appropriate points on the fabric. Texels will be made by integrating pieces of SMA wire into the fabric along grid lines, which will pull the textile when heated into their trained shape. Using small pieces of wire for each texel may also make the fabric feel less rigid.

The software to program the fabric’s texture aims to be as user-friendly as possible. As such, several different input styles were explored. First, an interface taking a single input photo of the desired texture was considered, with the photo serving as a height map. Another interface displays a visual library of smocked patterns which can be selected. Finally, a program was

considered which would directly display a smocking grid to the user such that they can select individual lines to create their own pattern.

### **Subsystems**

All prototypes of the texel system were created with the same synthetic fabric as their inactive material base. The system can be subdivided into smaller problems of actuators, electrical circuitry, components, material integration, and software.

### ***Actuator Selection and Training***

As discussed in the introduction, SMA was chosen as the preferred actuation method. Nitinol (50% nickel and 50% tin) was chosen as the specific alloy for its common availability and widely understood behavior as a shape change actuator even at the level of hobbyist projects. Selecting which Nitinol wire to use was a question of manufacturer documentation, gauge, and pulling force. While thicker wires would be able to pull more fabric with the same length, they heat less easily and add greater rigidity to the fabric. Thinner wires have opposite effects.

### **1mm Diameter Wire**

A generic brand of 1.0mm diameter Nitinol wire was tested first. The wire's activation temperature was listed as  $40\pm 2^{\circ}\text{C}$ . To train this wire, it was bent into the desired shape and held in place while being heated until red-hot, subsequently being quenched in room temperature water.

## Shape Language

Three activated shapes were tested with this wire, all with the goal of being able to precisely control their shape from flat to actuated with temperature monitoring and responsive current supply.

The first shape was a straight section of wire running as long as one texel with a looped eyelet section at the end. The straight section was parallel with the eyelet at rest and trained to be fully vertical when actuated. Thus, it was designed to pull connected fabric up to varying heights. The eyelet section was only used to affix the wire to the fabric base securely, remaining the same shape at rest and when heated. When several of these wires were placed in a line, however, the eyelet section alone proved insufficient to maintain an expected shape. Once one wire was in its actuated state, the wire next to it could drop and affect the overall slope unpredictably rather than staying at its current height and continuing to pull the fabric higher.

To maintain a consistent slope among adjacent wires, an 'L' shape was tested next. Like the previous iteration, a texel length of wire could bend from 0 to 90 degrees. The eyelet, however, was replaced with a consistently flat section of wire that was half the length of one texel. This section would overlap with the wire before it to provide support and keep a consistent slope from one wire to the next. As multiple wires in a row are activated, however, the fabric eventually collapses under their weight.

One final shape was tested to address the issue of counter-support. The 'L' shape was slightly modified such that both ends of the wire were half of one texel in length. The wires would no longer overlap with adjacent texels. Finally, the wire was positioned on the fabric so

that the bend in the center would face downward, turning it into an arch. As a result, heightened points along the fabric had enough support on either side to be sustained. This shape would also concede the ability to compound in height across multiple texels, with the height reached by any curve in each texel unit being limited to half of one texel length. Despite these limitations, the landscape of replicable textures was still vast.

### **Insights**

Explorations of shape language with this 1.0mm wire were concluded in favor of a thinner wire for several reasons. Although this gauge of wire produced more than enough pulling force to manipulate the base fabric, it added an unfavorable level of bulkiness and rigidity to the project. Its lower resistance would also make it more difficult to eventually actuate via joule heating as opposed to external heating elements. Additionally, as it was a generic brand, there was limited documentation from the supplier on its exact properties.

#### **0.127mm Diameter Wire**

0.127mm diameter Nitinol wire was chosen as a thinner alternative. Although the wire was branded as Muscle Wire, in-depth documentation was supplied via a Flexinol datasheet as a close approximation. To train this wire, it was bent into the desired shape, secured in place with copper tape, and heated to 500°C for five minutes before being quenched in room temperature water.

## Shape Language

This threadlike wire was first trained into an arch as previous experimentation suggested. However, it was unable to pull the fabric to a full close in this configuration. A spring shape was selected instead to smock the fabric more effectively using the same length.



Figure 3 – All attempted Nitinol shapes (at rest above, actuated below). Bending straight upwards with an eyelet for sewing (A), bending at a 90 degree angle with flat ends (B), bending into a downward arch (C), and coiling into a spring (D).

## Insights

This wire was preferred for further prototyping. Its higher resistance allowed for easier heating via electrical circuit, and it was small enough to have an insignificant effect on the flexibility and softness of the base fabric. Discrepancies in pulling force were later mitigated by adding length to the wire subunits, which made minimal difference to the fabric in any state. At rest, the thread was thin enough to go unnoticed, and while actuated into a spring shape any extra length was significantly condensed.

## *Electrical Circuit Design*

### Single Sub-Textel Circuit

Before being able to connect an array of texels, we first needed to know how one wire could be reliably actuated with an electrical circuit.

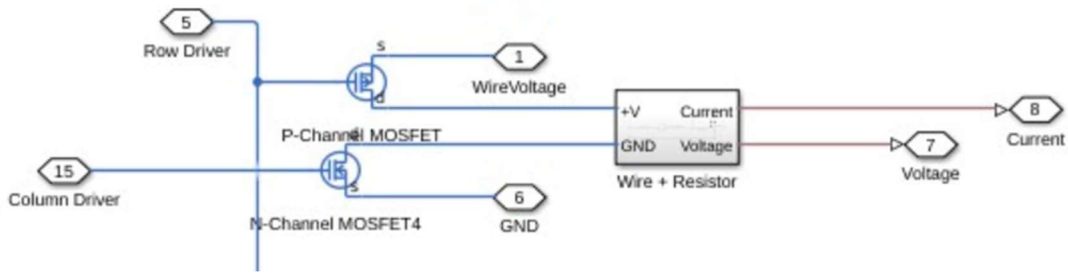


Figure 4– Single wire actuation circuit modeled on Simulink.

The first physical circuit implementation featured a thermistor for live temperature monitoring during actuation, attached to the nitinol wire. This thermistor added physical bulk on top of complexity to the circuit and control unit logic. Our aim was to monitor temperature in conjunction with actuation over repeated trials such that a reliable lookup table could be constructed with temperatures mapped to wire shape. If these values were predictable, it would allow for controlled actuation of the wire to some intermediate shape between its deactivated and activated states via precise heating. The desired temperature could be achieved by modulating the pulse width of the voltage sent through the circuit, and the thermistor could eventually be removed entirely. This plan suffered from three main limitations. Firstly, our thermistor could not give precise enough readings to decipher intermediary changes in the state of the wire. There was also a delay in the time that it took for the thermistor to register a temperature change and send this data to the main control unit. Varying atmospheric temperatures during use could make the calculated temperature mappings inaccurate, and sustaining a certain temperature in the wire would likely require live feedback from a thermistor to correct fluctuations. For these reasons, binary actuation states were preferred in subsequent prototypes over precise control. From this prototype, we were also able to note the current required to actuate the wire and verify the estimate from the datasheet.

## Texel Array Circuit

Upon verifying that one texel wire could be actuated, we then scaled up the circuit and introduced row and column drivers to actuate selected texels within an array.

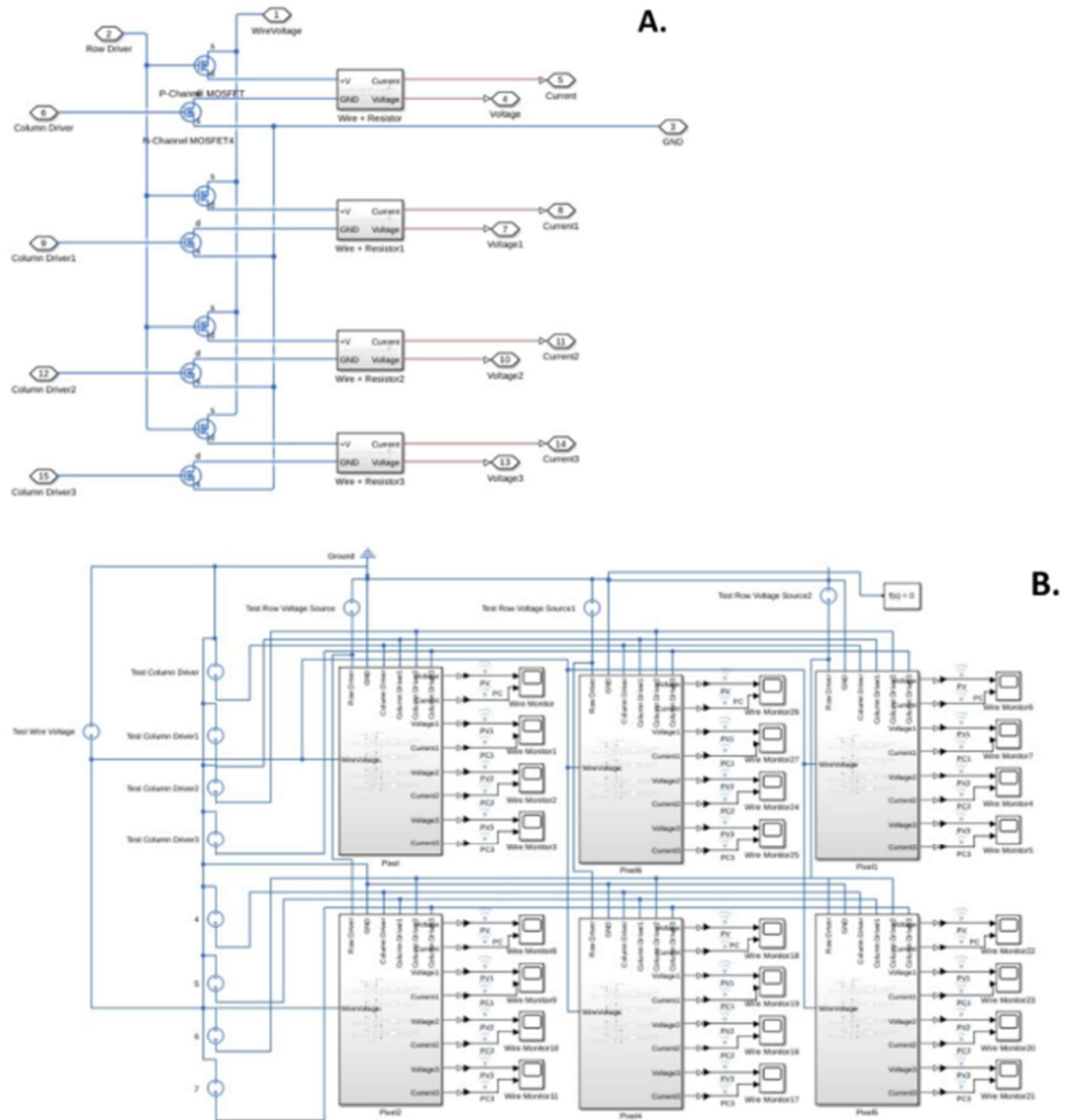


Figure 5– Four individual wires arranged into a texel unit (A), and an example circuit for a two-dimensional array of texels (B) modeled in Simulink.

Simulink was used to model the behavior of the circuit with varying voltages applied to its inputs. While applying voltage in different configurations, we checked that expected wires were receiving or not receiving power, and that those receiving power were drawing the same amount of current observed to activate our previous physical test. For the purposes of this simulation, generic N-channel and P-channel MOSFET models from the Simscape library were used and Nitinol wires were modeled as resistors. Resistivity was hard coded according to the length of the wire that would be used in the prototype and datasheet metrics. The control unit was not included in the Simulink model.

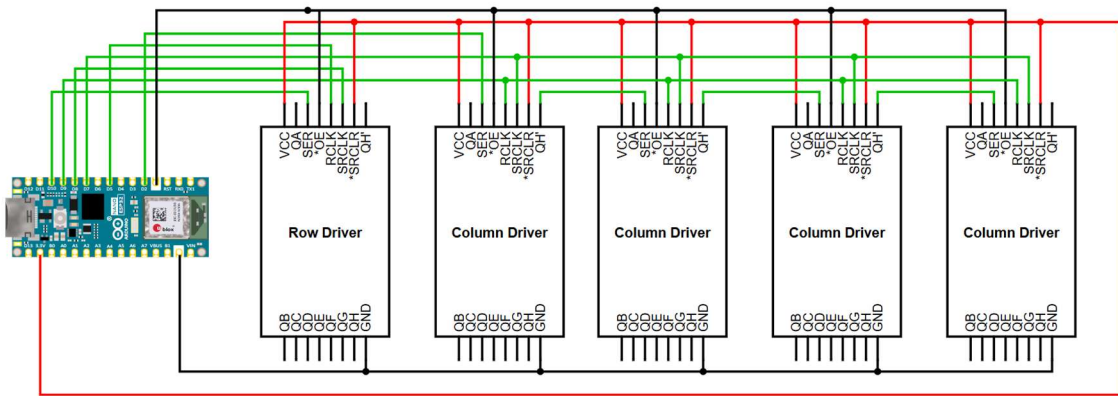


Figure 6– Control unit schematic. An Arduino ESP32 Nano connected to five shift registers. The first shift register serves as a row driver while the remaining four are chained together as column drivers.

An Arduino ESP32 Nano was chosen as the main control unit for ease in prototyping with its robust documentation, familiar programming suite, ample ports and additional features that could be of use when improving the prototype (e.g. Bluetooth connectivity). This main board would receive user input from the desktop software and determine which wires in each row would need to be activated. Similar to LCD screens using control boards and timer control units, it would coordinate impulses to open the gate of each row one by one before sending the

appropriate data to the texels in that row. Shift registers were used to expand the available output pins of the Arduino and act as the row and column drivers of LCD screens. One shift register controlled the gates of every row, and a chain of four additional shift registers had their outputs assigned to each column in a row. For the code used to program the control unit, see appendix B.

Power was supplied to the Arduino via wired connection to a laptop running the software interface. The Arduino then supplied power to the gates of each row. At first, the Arduino also supplied power to the texels via the column gates, however, as wire length increased to augment pulling force, more power was needed than the Arduino could supply. An external power supply was added to meet this need.

After the full circuit was theoretically sound, real parts would have to be chosen for their compatibility with one another and ability to carry out the use case.

### ***Electrical Component Selection***

Two MBA0204 resistors per wire were chosen to add the appropriate amount of resistance while dissipating the expected Power to be generated. BSR202N transistors on the low side and IRLML6401TRPBF transistors on the high side were chosen for their logic-level gate thresholds (low enough to be activated by an Arduino signal alone) and high available drain currents (due to Nitinol's large initial current draw). Small form factor was an important consideration for these components so as not to impede flexibility if sewn into the fabric. SNx4HC595 shift registers were selected for their appropriate operating voltage range. In addition, Adafruit conductive thread was used to form flexible connections within the fabric, and a variable benchtop power supply unit was used as the external power supply for the prototype.

## *Material Integration Techniques*

Given that much of the electronics would be on a fabric substrate, traditional components would have to be connected in a way that was flexible while remaining soundly connected. Multiple methods were explored to this effect, native either to fashion or traditional electronics.

### **Fashion Techniques**

The first major adaptation was connecting surface mount MOSFETs to other components either in mid-air or through fabric. To do this, conductive thread was wrapped around the metal lead of the component, which was then pressed into the side of the transistor to tighten the connection. The connection was further reinforced with a jeweler's crimp bead affixing both ends of the protruding conductive thread at the end of the lead. Finally, solder paste was melted with a heat gun to solidify the connection between lead and crimp bead. Knots were also used to create connections between components that were sewn in. Conductive threads could be directly tied together, or two parts could be affixed together at the same point on the fabric by sewing over and securing them with nonconductive thread. In any case, knots were covered with liquid adhesive to secure the bond and prevent fraying. Parts could be affixed to the fabric with a couching embroidery stitch, or by securing each end with a knot. Conductive thread could create 'traces' on both ends of the fabric by being sewn through in the desired direction, or they could be isolated to a single side if couched with nonconductive thread instead. Components were insulated from each other by layers of fabric or with a thin layer of adhesive.

## **Electrical Techniques**

Solder paste was used wherever rigidity could be tolerated. The selected Nitinol wire was not directly solderable, which presented an issue. To navigate this, rigid leads were bent around the wire with solder paste filling the resulting cavity. Once melted, the solder paste would entrap the wire in a solid connection. Plastic shrink tubing covered the connection for stability. Shrink tubing and pre-existing thin coatings around components were the main isolating factors.

## ***Software System***

The final component of a functioning prototype would be the ability to take input. Three types of software interfaces were considered for users to be able to program the fabric into patterns of their choosing.

## **Height Map Upload**

The first method was to take as input an image serving as a height map of the desired texture. With pixel values proportional to their height from a flat surface, a grid could be imposed onto the image such that intersecting lines on the grid would correspond with each physical texel in the fabric. This would be an important step especially when the resolution of pixels in the image and texels available do not exactly align. Along each line of the grid (representing one physical wire), the rate of change of the pixel values could be converted into an appropriate voltage to be sent to the wire, causing it to actuate to the desired slope. This method relies on the ability to precisely control actuation and thus would not work for a solution implementing binary wire states.

Although this interface would allow users to upload virtually any photo to attempt to replicate, the fabric itself would be limited in approximating many of these options. It may also require some basic understanding of concepts in computer vision or photo manipulation to ensure that an uploaded photo would serve as an adequate height map. Thus, unless the fabric technology were in a more advanced state, this style of interface would not present a clear and positive user experience. We opted not to program this interface for the present prototype due to these limitations.

### **Pattern Maker**

With this interface, the grid of texels is directly presented to the user. As such, expectations directly align with the capability of the device.

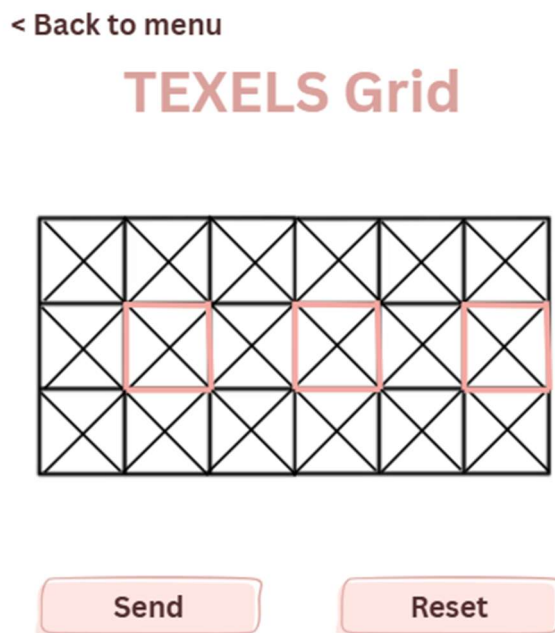


Figure 7 – Mockup of pattern-making software user interface.

Presented with the texel grid, users can select which lines they would like to smock together. The resulting design can be sent to the fabric or cleared. To recreate an existing smocking pattern, future users would be able to reference guides online or otherwise at their disposal. For the purposes of in-person testing and interviews, a guide document was created with three existing patterns and their grid layouts, viewable in appendix D. Although this option provides the most freedom for users to design any pattern that the fabric is capable of, it is not immediately intuitive to imagine what the result will look like based solely on the grid layout unless a user were to already be familiar with smocking and fabric manipulation.

### Library Selector



Figure 8– Mockup of library selector software user interface.

This interface presents images of smocked designs that the user can select from. It allows for a more intuitive process where the image on the screen reflects exactly how the fabric will arrange itself. Although it removes some freedom in being able to create custom patterns and select any available texel, future iterations could allow for users to upload their own patterns to be added to the library. This was the preferred interface for prototyping, and example code using pygame is available in appendix C.

# PROTOTYPING

## Preliminary Prototype

An initial prototype was assembled to test full integration of all subsystems. Subsequent limitations informed the development of the final version.

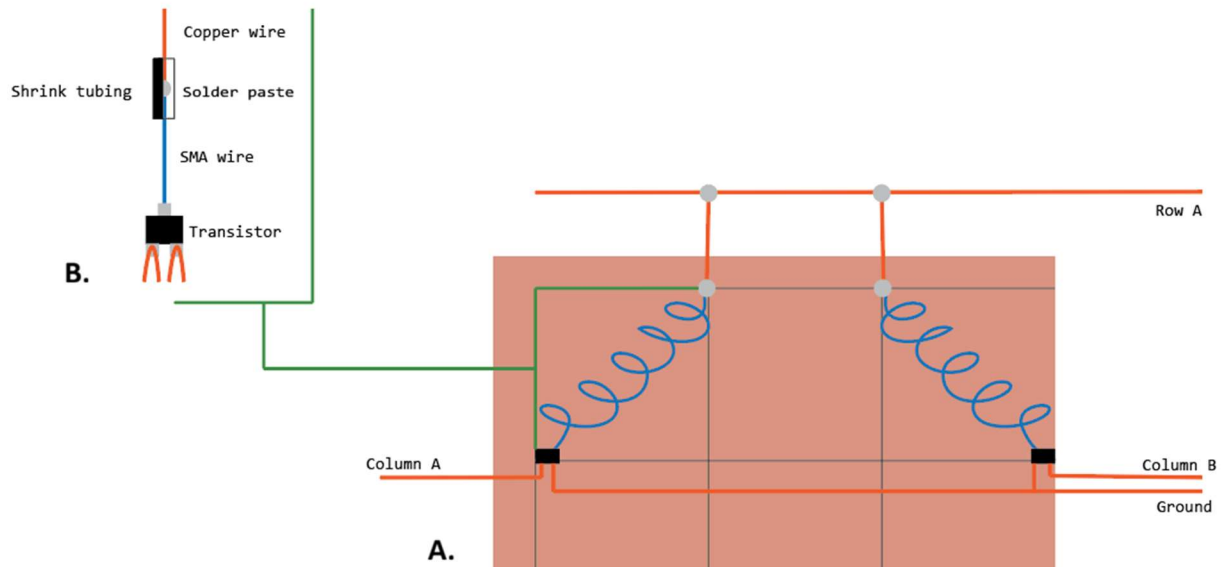


Figure 9– Schematic of initial prototype (A) with closeup of techniques used to connect each SMA wire (B).

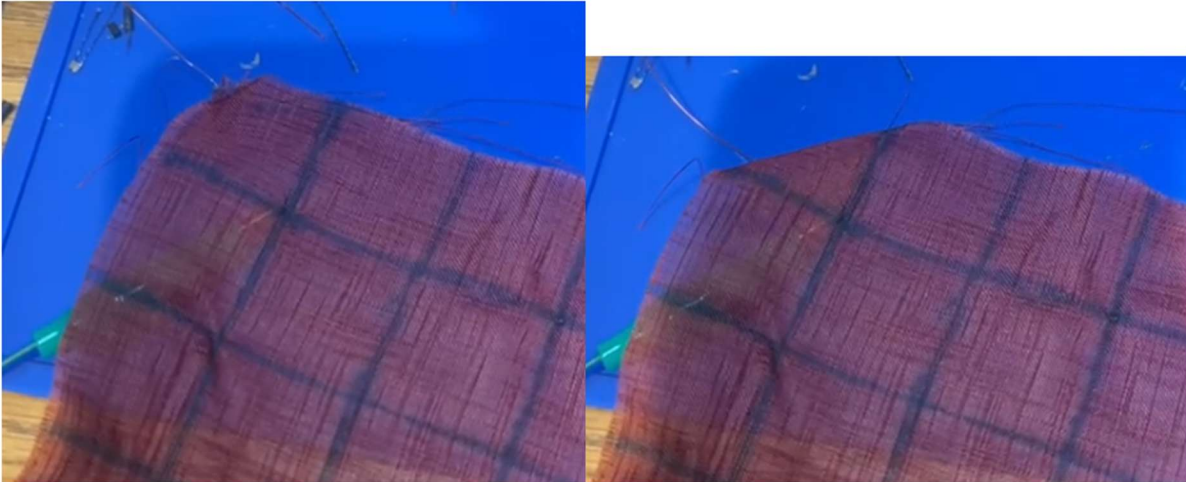


Figure 10– Testing of preliminary prototype. At rest (left) and 5 seconds after selecting actuation (right).

With only one layer of fabric, wires just long enough to connect two points on a flat grid square had sufficient pulling force to smock them together. As a result, the entire system could be powered by the 3.3V accessible from the Arduino without requiring external power supplies. The previously discussed circuitry and Arduino programming were employed. The software used for testing was a scaled-down version of that included in appendix C, with only a button to activate the “waves” pattern to verify movement.

The integration of electrical components proved to be the most limiting factor of this prototype. Solder paste and shrink tubing were used to make and reinforce connections. This resulted in small sections of the circuit that were rigid. Although the copper wire was thin, it was still a conspicuous addition to the fabric. It was also found to snap if bent repeatedly in the same place. The coating around the wire was helpful in keeping each one isolated as they crowded together towards the endpoints of the circuit, but it also had to be stripped off before making any connection. With this setup, the integration of each SMA wire into the grid would take several

hours by hand. Additionally, diagonal wires crossing over each other within a grid square could activate each other accidentally if not separated.

All of these factors in combination inspired the design of a new prototype iteration that would increase production efficiency by relying on techniques native to sewing rather than traditional electronics.

## Final Design

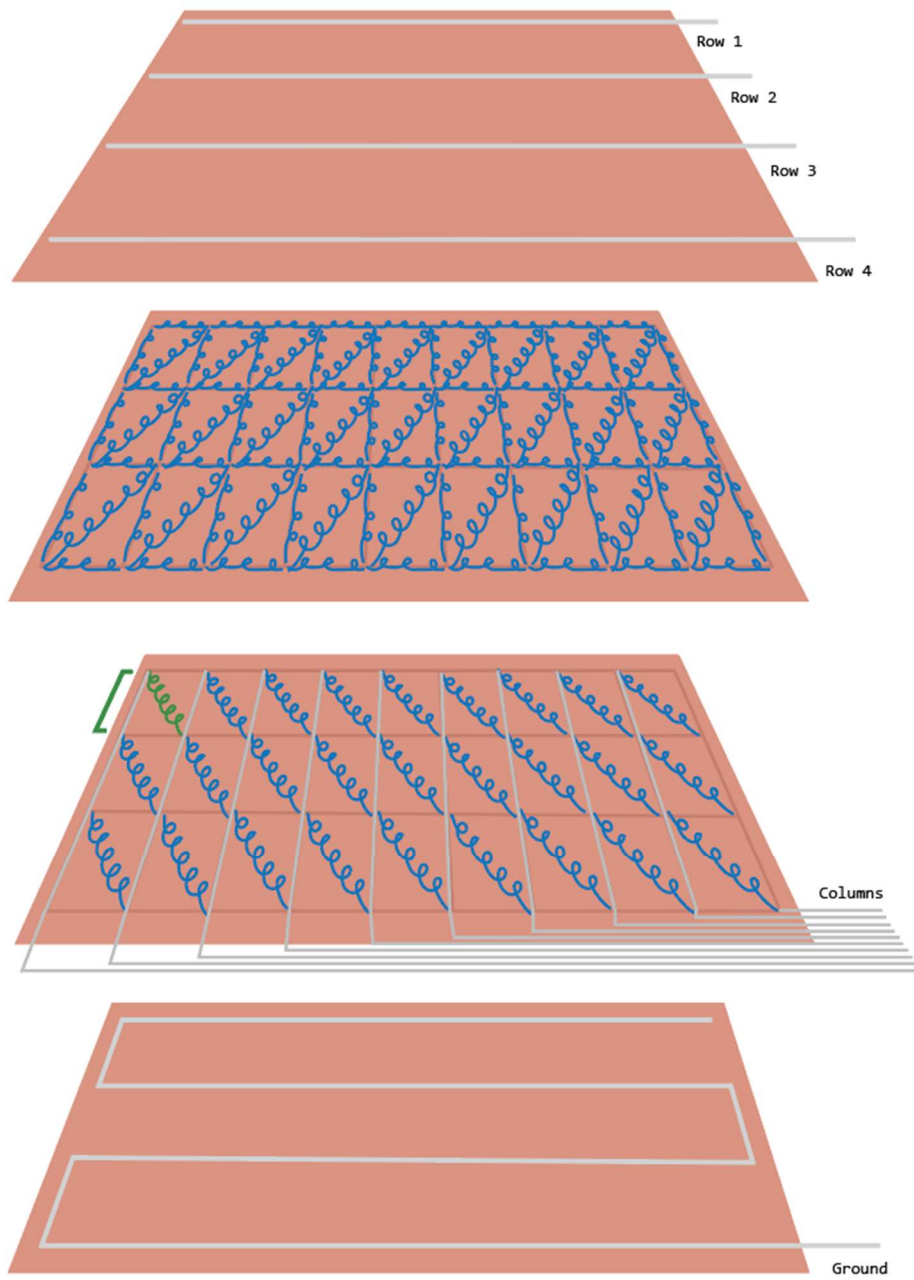


Figure 11– Schematic of the final layered design.

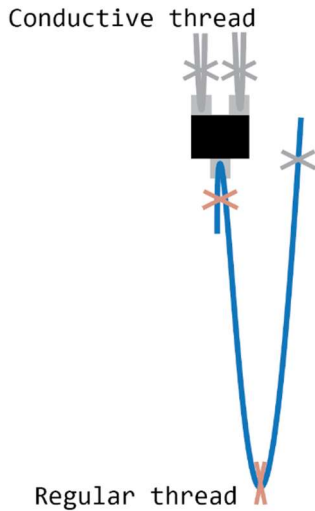


Figure 12– Closeup of sewn connections of one Nitinol wire.

The figures above describe the theory behind the second prototype iteration. First, overlapping wires were separated by a layer of fabric to prevent accidental activation. Because the design is multilayered, more pulling force is required to smock together two points on the grid. Thus, the length of SMA subunits was doubled, with the wire folded in half along each line on the grid. An external power supply was used to send an increased voltage through the row drivers to actuate this longer wire. Aside from this augmentation, the rest of the circuit remained the same. This design could be used with the fully developed software found in appendix C containing multiple patterns. This time, mostly fashion-oriented integration techniques were used, with layering for isolation rather than coatings or tubing, and conductive thread affixed to the fabric as opposed to copper wires. Conductive thread “traces” (like those illustrated for Rows or Ground) were sewn onto one side of the fabric using a couching stitch and regular thread such that they would not pass through and contact a component on another layer. Connections between components on different layers were achieved by sewing conductive thread through the layers and tying knots. These changes made it faster and easier to assemble the prototype, while

granting the device greater flexibility and softness overall. Flexible connection styles were more secure in the case of a moving fabric than rigid ones.

### Implementation

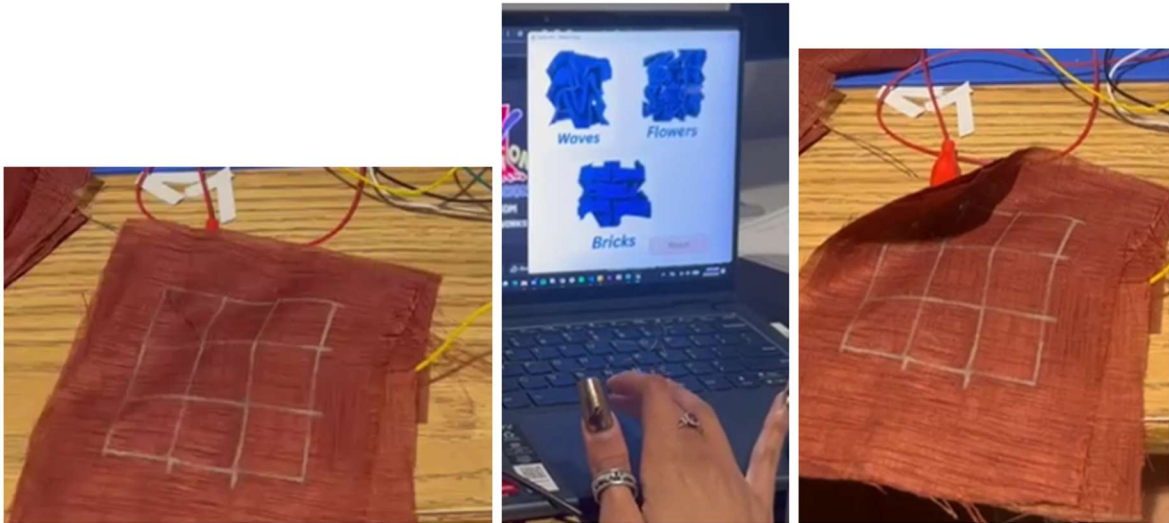


Figure 13– Testing of the layered iteration. At rest (left), engaging with interface to actuate (middle), and 3 seconds after selection (right).



Figure 14– Closeup of sewn connections implemented physically.

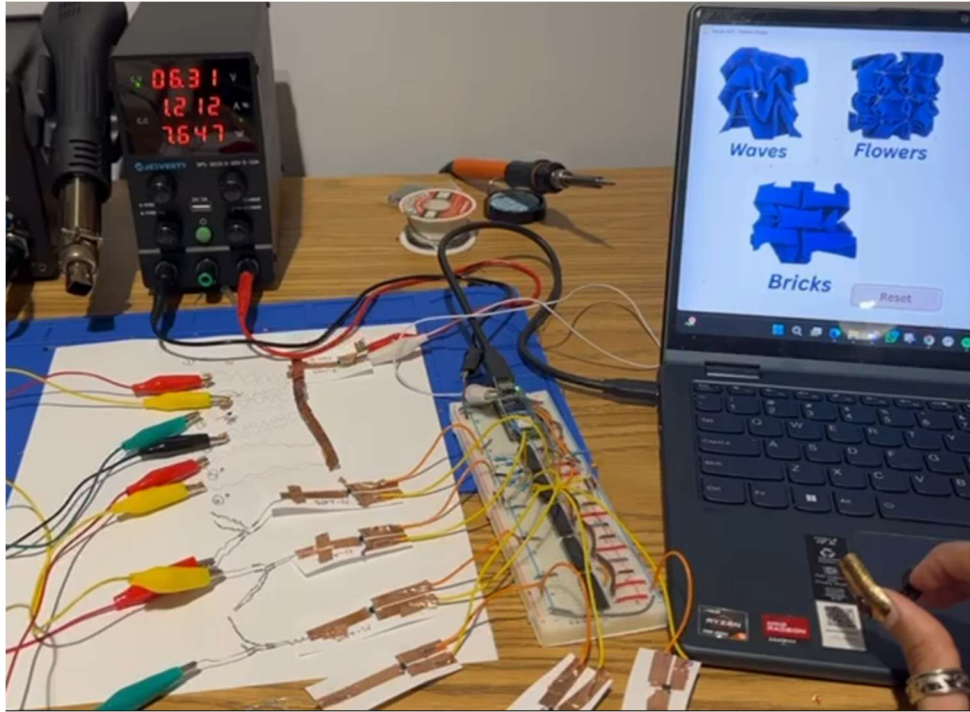


Figure 15 - Wires 1-4 activated for wave pattern.

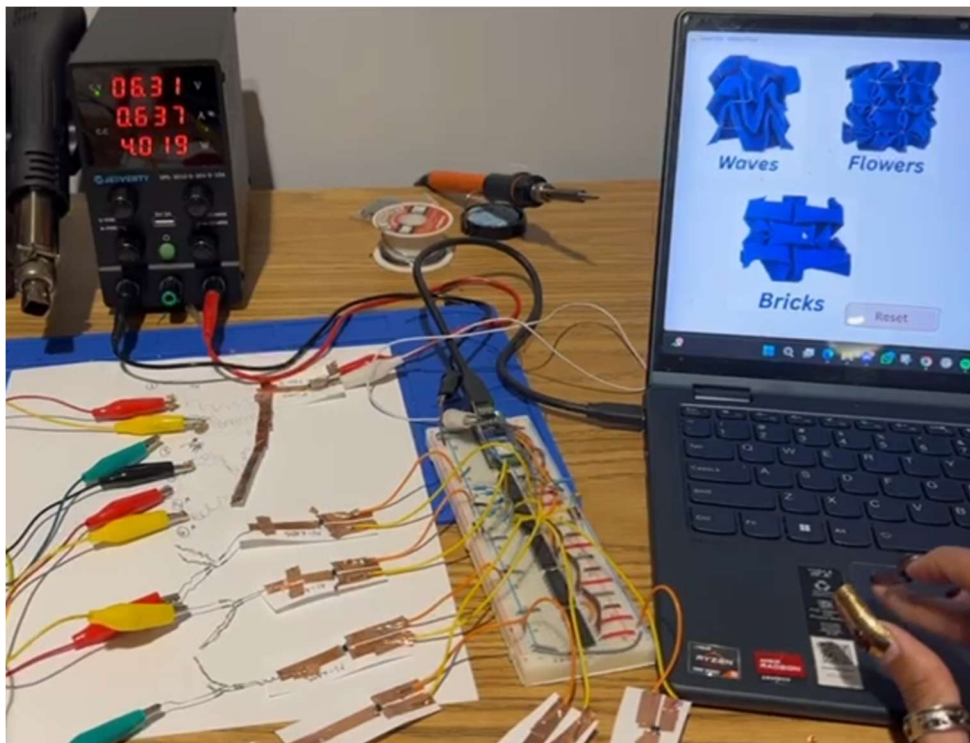


Figure 16 - Wires 2, 5, 6 activated for brick pattern.

Due to constraints in time and available materials, a subsection of the final design was tested. Sewn integration was confirmed to be responsive, and appropriate signals were sent depending on the selected pattern. More wires would have to be sewn in with additional time to recreate a fully visible pattern. For demonstration purposes, the software was altered to send continuous power to activated texels rather than looping through rows one by one. Videos are available via appendix E.

While simple patterns (such as waves) can be replicated accurately, those relying on tight connections (such as bricks) would appear looser than their traditional counterparts. This is because the SMA wire contracts into a shorter spring rather than pulling the fabric points completely together. Having to pull multiple layers of fabric also fatigues the trained shape gradually over repeated activations.

## DISCUSSION

### **Strengths of Current Prototype**

This system uses a simple universal grid layout to replicate any North American smocking pattern, making for a high number of possible outputs with low relative complexity. The fabrication methods employed maintain the fluidity of the original fabric. Connections between components remain strong through the deformation and movement of the fabric, and do not add detectable bulk to the surface. Assembly does not require the use of uncommon or complex machinery, and components are readily available for purchase by consumers at accessible prices. The interface is easy to interact with, requiring no prior knowledge of fashion or engineering techniques. Aside from current wired connections, the device could be reasonably sewn alongside traditional fabrics as part of a larger project.

### **Limitations and Recommendations for Further Development**

Along with these positive aspects, the current implementation of the prototype faces limitations. For example, the control unit software creates a shape by cycling through each row and sending a signal to activated wires only once. While this is enough to activate the desired wires in a row, they do not hold this heat and remain in their activated state for more than a few seconds after subsequent rows are selected. This could be rectified by repeating the main loop in the software until reset and introducing capacitors for each wire in the circuit, much like the standard LCD screen. Although the range of potential North American smocking patterns is diverse, precisely controlled actuation of the SMA would allow for nearly any texture to be replicated at a fixed resolution. Fatigue of Nitinol wires over time limits device usability but

would occur less noticeably in wires with greater pulling force. At present, the weight of the fabric layers is enough to mostly flatten the fabric swatch without human intervention when the “reset” button is pressed, but some additional smoothing may be required before selecting a new shape. Alternatively, counter-wires could be added which are trained into a flat position and activated when the shape is reset. Actuation takes several seconds, which may not be expected by users accustomed to modern digital user interfaces. Some indication of loading could be added to the software interface to lessen confusion or unease. In general, the software could be made more responsive with interaction conventions such as multiple states for buttons. In the future, it could be beneficial to add a way for users to upload their own patterns to the library. Currently, the device needs to be connected to a computer to run, but a Bluetooth connection is built into the ESP32 and could be set up with additional revisions to the software and receiver components. The external power supply could be replaced with battery power, though great care should be taken in making this revision to ensure that power is never drawn from the batteries while the rest of the control unit is off. Otherwise, the transistors could be damaged. Although the final design is much more efficient to assemble, it still takes time and attention to small details. Although readily available for purchase, shipping electrical components is another limitation on time. Looking to future use-cases, the current prototype was not wash-tested, although components that may be water-sensitive are coated in liquid adhesive. Waterproof components or a robust coating process would be preferred for later iterations. A custom PCB would also make connections more straightforward, as traces leading to the fabric could align with where they naturally land. A polyimide substrate would further supplement the flexibility of the fabric.

## **Future Evaluation and Applications**

Upon completion of a robust prototype with at least three full patterns, a user study could be conducted to gain insight into device usability and imagined use-cases. A study design was approved by the IRB consisting of interactive interviews with UCF students of varying majors. The interviews are to be conducted in three parts: first, guided interaction teaches students how the device is used and determines their ability to use it in response to specific instructions or desired patterns. Then, students are given free reign to explore the interface for a set amount of time. Finally, semi-structured interviews are conducted to understand their experience and ideas about applications within their field. Structured notes from interview sessions will be analyzed using an inductive coding approach to identify recurring patterns across participants. Codes will be iteratively refined and grouped into higher-level themes corresponding to each of the following research questions. A coding matrix will be maintained to support transparency and traceability of theme development.

- RQ1 - How do users perceive the intuitiveness and ease of use of the system during initial interaction? (codes related to confusion, learning curve, discoverability, frustration, flow)
- RQ2 - How do users compare and evaluate the library-based interface versus the grid-based interface, and in what contexts do they prefer each? (codes related to comparison, control, creativity, efficiency, context)
- RQ3 - What potential applications and future use cases do users envision for the system? (codes related to domains of application (fashion, accessibility, etc.))

For the IRB approval letter, see appendix A.

The prototype could also be used to create demos of its imagined use-cases in various fields. For example, it could form part of an expressive garment or connect to a Virtual Reality experience to simulate the surfaces of different objects. If the device is deemed usable enough to scale, it would be beneficial to thoroughly assess its sustainability either by ensuring it can withstand a long life-cycle of uses, or allowing it to degrade or be recycled after a shorter lifespan.

## CONCLUSION

This thesis proposed a novel shape-changing interface through the design and assembly of an active textile. Any traditional textile could be used as its base, with shape memory alloy wire serving as the active material of the system. Electricity was chosen to heat the active material, allowing for a self-contained system. In response to this activation energy, the textile employs North American smocking as its mode of communicating a variety of texture patterns. To control the actuation of this textile, we implement an electrical circuit that mirrors the functionality of an LCD's control unit in its ability to address individual segments of a two-dimensional array. The circuit is implemented physically using small components and sewn connections to preserve the fluidity of the original textile.

The proposed design creates a textile interface that is (1) Capable of recreating any North American smocking pattern; (2) Able to be reset and reactivated into different patterns multiple times; (3) Simple to interact with by selecting texture images in the associated software; (4) Accessible to build with common components and household sewing techniques.

Several approaches to actuator training, circuit design, components, integration techniques, and software were explored before complete assembly of the system. An initial prototype gave insight into final design recommendations that relied more heavily on traditional sewing techniques. These techniques were confirmed to maintain strong connections while allowing fluid movement of fabric, and the electrical circuit was confirmed to send appropriate signals based on pattern selection. With more wires integrated into a larger grid, full patterns would be visible to users. Improvements to the system discussed in the previous chapter would further enhance user experience.

**APPENDIX A:**

**IRB APPROVAL LETTER**



**Institutional Review Board**

FWA00000351  
IRB00001138, IRB00012110  
Office of Research  
3100 Technology Parkway,  
Suite 201

UNIVERSITY OF CENTRAL FLORIDA  
Orlando, FL 32826-3246

EXEMPTION DETERMINATION

March 26, 2026

Dear Joseph Laviola II:

On 3/26/2026, the IRB determined the following submission to be human subjects research that is exempt from regulation:

Type of Review:	Initial Study, Category 3(i)(B)
Title:	TEXELS: A Programmable Textile Interface for Replicating Textures
Investigator:	Joseph Laviola II
IRB ID:	STUDY00008872
Funding:	None
Documents Reviewed:	<ul style="list-style-type: none"> <li>• HRP-254-consent.pdf, Category: Consent Form;</li> <li>• HRP-255 - FORM - Request for Exemption.docx, Category: IRB Protocol;</li> <li>• TEXELS_Email.docx, Category: Recruitment Materials;</li> <li>• TEXELS-Flyer.png, Category: Recruitment Materials;</li> <li>• TEXELS-Interview-Questions.docx, Category: Interview / Focus Questions;</li> <li>• TEXELS-Pattern-Document.pdf, Category: Test Instruments;</li> <li>• TEXELS-Prototype-Images.pdf, Category: Test Instruments</li> </ul>

This determination applies only to the activities described in the IRB submission and does not apply should any changes be made. If changes are made, and there are questions about whether these changes affect the exempt status of the human research, please submit a modification request to the IRB. Guidance on submitting Modifications and Administrative Check-in is detailed in the Investigator Manual (HRP-103), which can be found by navigating to the IRB Library within the IRB system. When you have completed your research, please submit a Study Closure request so that IRB records will be accurate. If you have any questions, please contact the UCF IRB at 407-823-2901 or [irb@ucf.edu](mailto:irb@ucf.edu). Please include your project title and IRB number in all correspondence with this office.

Sincerely,  
The UCF IRB

## **APPENDIX B:**

### **ARDUINO CODE**

```

// Pin setup-----
//Gate Shift pin 12

const int GlatchPin = D5;

//^ pin 11

const int GclockPin = D8;

//^ ds pin 14

const int GdataPin = D10;

//Source Shift pin 12

const int SlatchPin = D9;

//^ pin 11

const int SclockPin = D7;

//^ ds pin 14

const int SdataPin = D2;

const byte GStartByte = 0b00000001;

//-----

int sourceOutputs[7];

// HELPERS -----

void write32(int val){

    shiftOut(SdataPin, SclockPin, MSBFIRST, (val >> 24));

```

```

    shiftOut(SdataPin, SclockPin, MSBFIRST, (val >> 16));
    shiftOut(SdataPin, SclockPin, MSBFIRST, (val >> 8));
    shiftOut(SdataPin, SclockPin, MSBFIRST, val);
}

//-----

void setup() {
    pinMode(GlatchPin, OUTPUT);
    pinMode(GclockPin, OUTPUT);
    pinMode(GdataPin, OUTPUT);

    pinMode(SlatchPin, OUTPUT);
    pinMode(SclockPin, OUTPUT);
    pinMode(SdataPin, OUTPUT);

    Serial.begin(9600);
}

void loop() {
    //Populate data only when a full pattern has been sent
    while (Serial.available() < 28);

    //Populate source outputs one row at a time

```

```

for (int i = 0; i < 7; i++){
    int rowvalue = 0;
    //Get int from 4 bytes
    for (int j = 0; j < 4; j++){
        rowvalue = (rowvalue << 8) | Serial.read();
    }
    sourceOutputs[i] = rowvalue;
}

//Activate pattern based on input
actuation();
}

// PRIORITY: ACTIVATE MULTIPLE TEXEL WIRES (connected to circuit) -----
void actuation(){
    for(int i = 0; i < 4; i++){
        //Turn on row of interest
        digitalWrite(GlatchPin, LOW);
        byte GCurrByte = ~(GStartByte << i);
        shiftOut(GdataPin, GclockPin, MSBFIRST, GCurrByte);
        digitalWrite(GlatchPin, HIGH);

        //Send source values

```

```
digitalWrite(SlatchPin, LOW);  
write32(sourceOutputs[i]);  
digitalWrite(SlatchPin, HIGH);  
delay(10000); //can experiment with how long it takes to actuate  
digitalWrite(SlatchPin, LOW);  
shiftOut(GdataPin, GclockPin, MSBFIRST, 0b11111111);  
digitalWrite(SlatchPin, HIGH);  
}  
}  
//-----
```

**APPENDIX C:**

**USER INTERFACE CODE**

```

import serial

import pygame

# SETUP -----

pygame.init()

comms = serial.Serial('COM12', 9600, writeTimeout=0)

CURRENT_PATTERN = []

SIZE = 700

screen = pygame.display.set_mode((SIZE, SIZE))

pygame.display.set_caption("Texels GUI - Pattern Picker")

#load button images

waves_img = pygame.image.load('wavesBtn.png').convert_alpha()

brick_img = pygame.image.load('brickBtn.png').convert_alpha()

flower_img = pygame.image.load('flowerBtn.png').convert_alpha()

off_img = pygame.image.load('offBtn.png').convert_alpha()

#-----

# PATTERN DATABASE (could be imported later) -----

WAVES_PATTERN = [0b00000010, 0b00000001, 0b00000010, 0b00000001,
                 0b00000000, 0b00000000, 0b00000000, 0b00000000,

```

0b00000010, 0b00000001, 0b00000010, 0b00000001,  
0b00000000, 0b00000000, 0b00000000, 0b00000000,  
0b00000010, 0b00000001, 0b00000010, 0b00000001,  
0b00000000, 0b00000000, 0b00000000, 0b00000000,  
0b00000010, 0b00000001, 0b00000010, 0b00000001]

BRICK\_PATTERN = [0b00000001, 0b00100110, 0b01010001, 0b00100000,  
0b00000100, 0b01000000, 0b00000100, 0b01000000,  
0b00000110, 0b01010001, 0b00100110, 0b01010000,  
0b00000000, 0b00000100, 0b01000000, 0b00000000,  
0b00000001, 0b00100110, 0b01010001, 0b00100000,  
0b00000100, 0b01000000, 0b00000100, 0b01000000,  
0b00000110, 0b01010001, 0b00100110, 0b01010000]

FLOWER\_PATTERN = [0b00000000, 0b00000000, 0b00000000, 0b00000000,  
0b00000000, 0b11001000, 0b11001000, 0b11001000,  
0b00000000, 0b01000000, 0b01000000, 0b01000000,  
0b00000000, 0b11001000, 0b11001000, 0b11001000,  
0b00000000, 0b01000000, 0b01000000, 0b01000000,  
0b00000000, 0b11001000, 0b11001000, 0b11001000,  
0b00000000, 0b01000000, 0b01000000, 0b01000000]

OFF = [0b00000000, 0b00000000, 0b00000000, 0b00000000,

```
0b00000000, 0b00000000, 0b00000000, 0b00000000,  
0b00000000, 0b00000000, 0b00000000, 0b00000000,  
0b00000000, 0b00000000, 0b00000000, 0b00000000,  
0b00000000, 0b00000000, 0b00000000, 0b00000000,  
0b00000000, 0b00000000, 0b00000000, 0b00000000,  
0b00000000, 0b00000000, 0b00000000, 0b00000000]
```

```
#-----
```

```
# HELPERS -----
```

```
#button class
```

```
class Button():
```

```
    def __init__(self, x, y, image, scale = 1):
```

```
        width = image.get_width()
```

```
        height = image.get_height()
```

```
        self.image = pygame.transform.scale(image, (int(width * scale), int(height * scale)))
```

```
        self.rect = self.image.get_rect()
```

```
        self.rect.topleft = (x, y)
```

```
        self.clicked = False
```

```
    def draw(self):
```

```
        action = False
```

```
        #get mouse position
```

```
        pos = pygame.mouse.get_pos()
```

```

#check mouseover and clicked conditions

if self.rect.collidepoint(pos):

    if pygame.mouse.get_pressed()[0] == 1 and self.clicked == False:

        self.clicked = True

        action = True

if pygame.mouse.get_pressed()[0] == 0:

    self.clicked = False

#draw button on screen

screen.blit(self.image, (self.rect.x, self.rect.y))

return action

#-----

# GAME -----

#setup

waves_button = Button(40, 20, waves_img, 0.5)

brick_button = Button(140, 350, brick_img, 0.5)

flower_button = Button(370, 20, flower_img, 0.45)

off_button = Button(400, 600, off_img, 0.3)

```

```
#loop

run = True

while run:

    screen.fill((250, 250, 250))

    if waves_button.draw():

        #send waves data via serial

        CURRENT_PATTERN = WAVES_PATTERN.copy()

        for byte in CURRENT_PATTERN:

            comms.write(bytes([byte]))

    if brick_button.draw():

        #send brick data via serial

        CURRENT_PATTERN = BRICK_PATTERN.copy()

        for byte in CURRENT_PATTERN:

            comms.write(bytes([byte]))

    if flower_button.draw():

        #send flower data via serial

        CURRENT_PATTERN = FLOWER_PATTERN.copy()

        for byte in CURRENT_PATTERN:

            comms.write(bytes([byte]))
```

```
if off_button.draw():  
    #send off data via serial  
    CURRENT_PATTERN = OFF.copy()  
    for byte in CURRENT_PATTERN:  
        comms.write(bytes([byte]))  
  
#event handler  
for event in pygame.event.get():  
    #quit game  
    if event.type == pygame.QUIT:  
        run = False  
  
pygame.display.update()  
  
comms.close()  
pygame.quit()
```

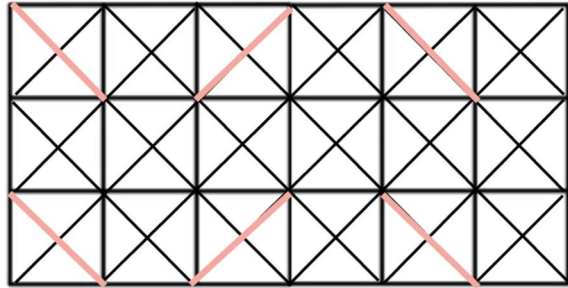
**APPENDIX D:**

**NORTH AMERICAN SWATCH SAMPLES**

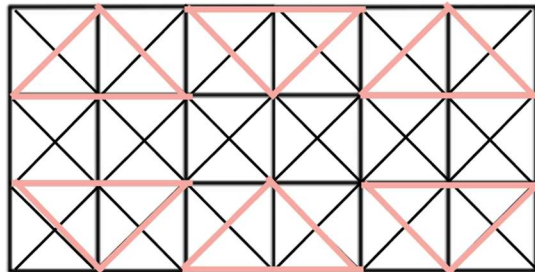
# NORTH AMERICAN SMOCKING PATTERNS

*North American Smocking* is a technique for creating textures on the surface of fabrics. The colored lines in the grids indicate which points on the fabric to tie together with thread, resulting in repeated deformations.

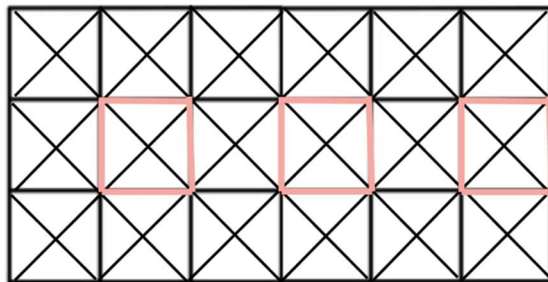
## 1. Wave Pattern



## 2. Brick Pattern



## 3. Flower Pattern



## **APPENDIX E:**

### **VIDEOS AND ADDITIONAL MEDIA OF PROTOTYPES**

Full videos are available to view at  
<https://tinyurl.com/TEXELSMedia>

## REFERENCES

- [1] P. G. Tortora, *Dress, fashion, and technology : from prehistory to the present*. London ; New York: Bloomsbury, 2015.
- [2] S. Tibbits, *Active matter*. Cambridge, Massachusetts: The Mit Press, 2017.
- [3] M. L. Rivera, M. Moukperian, D. Ashbrook, J. Mankoff, and S. E. Hudson, "Stretching the Bounds of 3D Printing with Embedded Textiles," *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, May 2017, doi: <https://doi.org/10.1145/3025453.3025460>.
- [4] A. Abouraddy, J. Kaufman, M. Monroe, and F. Tan, "Color-changing fabric and applications," May 07, 2024 Accessed: Sep. 22, 2024. [Online]. Available: <https://imageppubs.uspto.gov/dirsearch-public/print/downloadPdf/11976389>
- [5] J. Forman *et al.*, "FibeRobo: Fabricating 4D Fiber Interfaces by Continuous Drawing of Temperature Tunable Liquid Crystal Elastomers," Oct. 2023, doi: <https://doi.org/10.1145/3586183.3606732>.
- [6] J. H. (Heather) Kim, K. Huang, S. White, M. Conroy, and C. H.-L. Kao, "KnitDermis: Fabricating Tactile On-Body Interfaces Through Machine Knitting," *Designing Interactive Systems Conference 2021*, Jun. 2021, doi: <https://doi.org/10.1145/3461778.3462007>.
- [7] L. Buechley, M. Eisenberg, J. Catchen, and A. Crockett, "The LilyPad Arduino," *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, 2008, doi: <https://doi.org/10.1145/1357054.1357123>.
- [8] H. Peng, J. Mankoff, S. E. Hudson, and J. C. McCann, "A Layered Fabric 3D Printer for Soft Interactive Objects," Apr. 2015, doi: <https://doi.org/10.1145/2702123.2702327>.
- [9] F. Davis, A. Roseway, E. Carroll, and M. Czerwinski, "Actuating mood," *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction - TEI '13*, 2013, doi: <https://doi.org/10.1145/2460625.2460640>.
- [10] J. Forman, T. Tabb, Y. Do, M.-H. Yeh, A. Galvin, and L. Yao, "ModiFiber," May 2019, doi: <https://doi.org/10.1145/3290605.3300890>.
- [11] O. Kilic Afsar *et al.*, "OmniFiber: Integrated Fluidic Fiber Actuators for Weaving Movement based Interactions into the 'Fabric of Everyday Life,'" The 34th Annual ACM Symposium on User Interface Software and Technology, Oct. 2021, doi: <https://doi.org/10.1145/3472749.3474802>.
- [12] A. Wakita, A. Nakano, and M. Ueno, "SMAAD Surface: A tangible interface for smart material aided architectural design," *Proceedings of the International Conference on Computer-Aided Architectural Design Research in Asia*, Jan. 2011, doi: <https://doi.org/10.52842/conf.caadria.2011.355>.
- [13] K. Nakagaki, A. Dementyev, S. Follmer, J. Paradiso, and H. Ishii, "ChainFORM: A Linear Integrated Modular Hardware System for Shape Changing Interfaces," vol. 16, 2016, doi: <https://doi.org/10.1145/2984511.2984587>.
- [14] M. L. Rivera, J. Forman, S. E. Hudson, and L. Yao, "Hydrogel-Textile Composites," *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, Apr. 2020, doi: <https://doi.org/10.1145/3334480.3382788>.
- [15] H. Takahashi and J. Kim, "3D Printed Fabric," *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, Oct. 2019, doi: <https://doi.org/10.1145/3332165.3347896>.

- [16] Masaru Ohkubo and T. Nojima, “SmartFiber,” Jan. 2018, doi: <https://doi.org/10.1145/3174910.3174949>.
- [17] Y. Tahouni, I. P. S. Qamar, and S. Mueller, “NURBSforms,” Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction, Feb. 2020, doi: <https://doi.org/10.1145/3374920.3374927>.
- [18] Ahmed Amine Chafik, J. Gaber, Souad Tayane, M. Ennaji, J. Bourgeois, and Tarek El Ghazawi, “From conventional to programmable matter systems: A review of design, materials, and technologies,” *ACM Computing Surveys*, vol. 56, no. 8, pp. 1–26, Apr. 2024, doi: <https://doi.org/10.1145/3653671>.
- [19] A. Parkes and H. Ishii, “Bosu,” Aug. 2010, doi: <https://doi.org/10.1145/1858171.1858205>.
- [20] E. Hawkes et al., “Programmable matter by folding,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 28, pp. 12441–12445, Jun. 2010, doi: <https://doi.org/10.1073/pnas.0914069107>.
- [21] Ahmed Amine Chafik, J. Gaber, Souad Tayane, and M. Ennaji, “Programmable smart articulated interface,” HAL (Le Centre pour la Communication Scientifique Directe), Jul. 2022, doi: <https://doi.org/10.1109/hsi55341.2022.9869504>.
- [22] S. Kamrava, D. Mousanezhad, H. Ebrahimi, R. Ghosh, and A. Vaziri, “Origami-based cellular metamaterial with auxetic, bistable, and self-locking properties,” *Scientific Reports*, vol. 7, no. 1, Apr. 2017, doi: <https://doi.org/10.1038/srep46046>.
- [23] S. Kamrava, R. Ghosh, Z. Wang, and A. Vaziri, “Origami-Inspired Cellular Metamaterial With Anisotropic Multi-Stability,” *Advanced Engineering Materials*, vol. 21, no. 2, p. 1800895, Oct. 2018, doi: <https://doi.org/10.1002/adem.201800895>.
- [24] S. Kamrava, R. Ghosh, Y. Yang, and A. Vaziri, “Slender origami with complex 3D folding shapes,” *EPL (Europhysics Letters)*, vol. 124, no. 5, p. 58001, Dec. 2018, doi: <https://doi.org/10.1209/0295-5075/124/58001>.
- [25] A. Buscicchio, “Textile-based deployable structures and robotic systems for aerospace applications,” Politecnico di Bari, 2024.
- [26] E. Demaine and T. Tachi, “Origamizer: A Practical Algorithm for Folding Any Polyhedron,” no. 34, pp. 34–35, 2017, doi: <https://doi.org/10.4230/LIPIcs.SoCG.2017.34>.
- [27] M. Kshirsagar and B. Kandasubramanian, “Origami fabrication techniques for enhanced fiber reinforced composites: A review,” *Hybrid Advances*, vol. 7, p. 100274, Dec. 2024, doi: <https://doi.org/10.1016/j.hybadv.2024.100274>.
- [28] C. Wang, C. Yang, J. Li, and M. Zhu, “Research on the Creative Application of Origami Performance Techniques in Clothing,” *Fibres & Textiles in Eastern Europe*, vol. 30, no. 4, pp. 43–53, Jul. 2022, doi: <https://doi.org/10.2478/ftce-2022-0035>.
- [29] L. Sun et al., “Stimulus-responsive shape memory materials: A review,” *Materials & Design*, vol. 33, pp. 577–640, Jan. 2012, doi: <https://doi.org/10.1016/j.matdes.2011.04.065>.
- [30] *Encyclopedia of Smart Materials*. Elsevier, 2022.
- [31] M. Kim et al., “Shape Memory Alloy (SMA) Actuators: The Role of Material, Form, and Scaling Effects,” *Advanced Materials*, Apr. 2023, doi: <https://doi.org/10.1002/adma.202208517>.
- [32] Y. Zhou et al., “Magnetic properties of smart textile fabrics through a coating method with NdFeB flake-like microparticles,” *Journal of Engineered Fibers and Fabrics*, vol. 14, p. 155892501986570, Jan. 2019, doi: <https://doi.org/10.1177/1558925019865708>.
- [33] N. Benbernou, E. D. Demaine, M. L. Demaine, and Aviv Ovadya, “A Universal Crease Pattern for Folding Orthogonal Shapes,” *arXiv (Cornell University)*, Jan. 2009, doi: <https://doi.org/10.48550/arxiv.0909.5388>.

- [34] Behnaz Farahi, "Caress of the Gaze: A Gaze Actuated 3D Printed Body Architecture," Jan. 2016, doi: <https://doi.org/10.52842/conf.acadia.2016.352>.
- [35] J. Alexander et al., "Grand Challenges in Shape-Changing Interface Research," Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18, 2018, doi: <https://doi.org/10.1145/3173574.3173873>.
- [36] J. R. Glover, "A theoretical design model using the shape memory alloy, nitinol, in a robotic finger," 2022. <https://scholarworks.calstate.edu/concern/theses/m613n487h>.
- [37] I. Pappas, S. Siskos, and C. A., "Active-Matrix Liquid Crystal Displays - Operation, Electronics and Analog Circuits Design," New Developments in Liquid Crystals, Nov. 2009, doi: <https://doi.org/10.5772/9686>.
- [38] M.-K. Chang, S. Jeong, D. Kim, and H. Nam, "Review of Integrated Gate Driver Circuits in Active Matrix Thin-Film Transistor Display Panels," Micromachines, vol. 15, no. 7, p. 823, Jun. 2024, doi: <https://doi.org/10.3390/mi15070823>.
- [39] C. Wolff, The art of manipulating fabric. Iola, Wis.: Krause, [20]14, 1996.