“THIS = THEN = THAT”

PROTOTYPE

Professor: Elio Bidinost & Sabine Rosenberg

Course: CART 360

Student: Liu WenYue (40098425)

Date: November 6th, 2019

Github repository URL: https://github.com/Liu-WenYue/cart360-2019

Why Do We Prototype & Fidelity Levels

Prototyping is an essential process for bringing a concept idea to life. It gives us an opportunity to understand what the project going to do and going to be. In the process of making a prototype, we start to break down our original ideas and do some experiments to see the actual feasibility of each of these ideas. When we are looking into these ideas in detail, we begin to look for a solution to make these ideas work in a systematic manner. By questioning ourself "do we need more research or supporting information to accomplish this part?", "what are the materials suitable for this project?" And later in high fidelity level, we may even ask ourselves if the interaction between the product and user is meaningful enough and if there are any possible improvements for the prototype. By exploring the ideas in-depth, we encounter problems and obstacles that we may not even expect during the concept idea stage. With the problems discovered in the prototyping stage, we have sufficient time to solve it and even carry out some user tests to see if the problems are successfully solved. Moreover, prototyping uses cheap and disposable materials that afford multiple failures. Last but not least, prototyping illustrates our ideas better than just pitch it out in words. People will get a clearer direction of what our project about, and therefore they can provide more valuable suggestions that concerned with the project.

According to Kathryn McElroy, there are three different fidelity levels during prototyping. Firstly, the low fidelity level prototyping is prototypes that can be made using the materials around us very quickly and easily. It always in the forms of paper prototypes, storyboards, and hand-drawn sketches. For my low fidelity level prototyping, I did some rough sketches that show the volume, design, and the desired material of the plushie. Moreover, I also drew some storyboards to illustrate the interactions between the plushie and its users.

Secondly, the mid-fidelity level prototyping refers to prototypes that begin to look like our final product. It starts to test the interactions that we want to have for our final product. This is the stage where we do experiments to refine our ideas from the previous stage. I started my mid-fidelity level prototyping by reviewing my concept idea and questioning myself, "Do I need more information or research for each of these parts?", "What kind of sensors should I use to track the users' input?", "What materials should I use for the plushie?" and "How to approach each interaction design in my storyboards?" With these questions in mind, I refined my interaction designs and started testing out different materials for the prototype.

My first material test is for the plushie's body. The first interaction between the users and the plushie will be the tactile experience when users first hold the plushie in their hands. After feeling a range of fabrics by holding in my own hands, I decided to go with this dark blue velvet fabric (The last material in Fg.1).

(Fg1. Material Tests)

To cooperate with the soft materiality of the plushie, I decided to use fabric sensors for tracking user inputs. The progress of making the pressure sensor and fabric button will be discussed more clearly in the next section. After finished making the sensors and making sure all the soldering is working properly, I started coding to test if my pressure sensor works with the mozzi audio synthesis library for Arduino.

Lastly, the high fidelity level prototyping refers to the prototype that is finished in terms of visual design and functionality. The code is error-free, and the desired interactions are all working well. This prototype is very close to the final artifact. It does the final user tests and allows us to make our final adjustments. At my high fidelity level prototyping, I will have code for all the possible interactions for my fabric sensors, and have the materials of the plushie covers these sensors and wires.

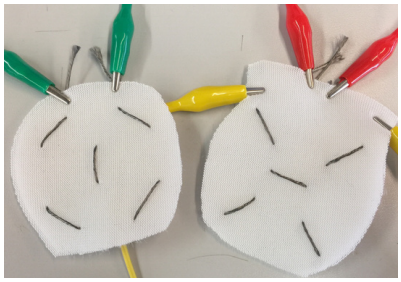
Sensors

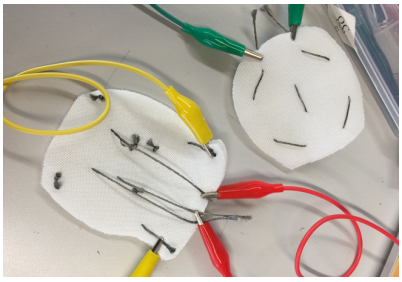
In this interactive plushie project, I will be mainly working with fabric sensors. I will be using the pressure sensor matrix for the stomach of my plushie. It tracks the tactile input from the user, and output audios depending on which area and how strong had the user just touched. Moreover, I will be adding fabric buttons on the plushie's ears. It also tracks the tactile input from the user, and output a blinking LED effect.

Pressure Sensor Matrix

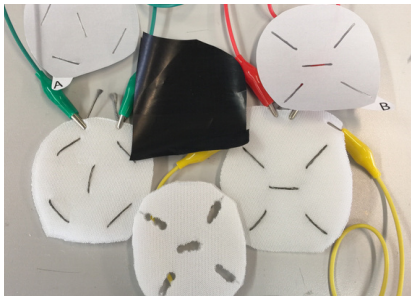
The pressure sensor matrix allows me to build a tactile sensor using just the soft materials. I am able to control the shape of the sensor and make variations to the design because of the flexibility of the soft fabrics. Moreover, I learned how do the sensors work during the process of actually making these sensors. For this pressure sensor matric, it allows me to track the analog input values (finger pressure from the users) from different areas of the interface (plushie's stomach).

The pressure sensor matrix is made of three layers of non-conductive fabrics. The top and bottom layers have conductive threads sewed on them. The sewing is in a special manner where the conductive threads on the two layers are in contact with each other (Fg2). The conductive thread on the top layer is one complete thread, whereas the threads on the bottom layer are five separate threads. It means every single thread is receiving a signal from the top layer, and sending it to the microcontroller. The middle layer has punched holes at the contacting area of the conductive threads, its thickness holds the space between the top and bottom layer (Fg3). When the users press the top layer, it gets contact with the bottom layer again and therefore completes the circuit.



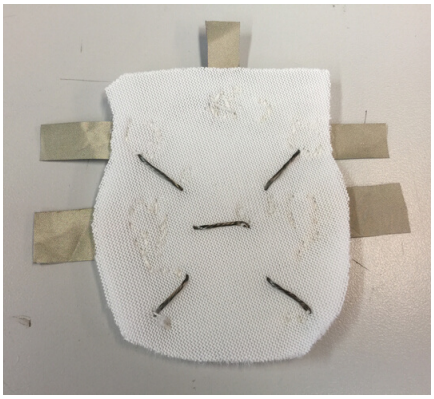


(Fg2. Arrangement of the thread)

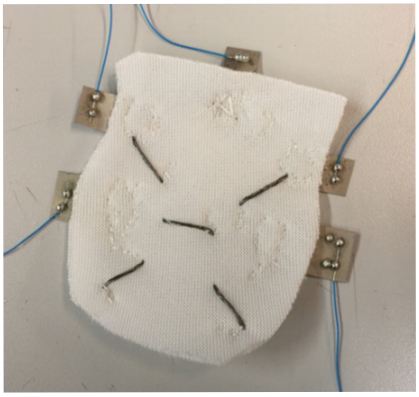


(Fg3. Material prepared)

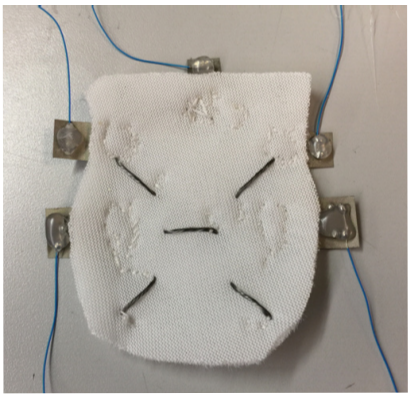
To connect each of the five conductive threads to the microcontroller, I need to solder wires on to a small piece of conductive fabric that covers the end of the threads. Moreover, I also applied some hot glue onto each of the soldering parts to protect the connection.



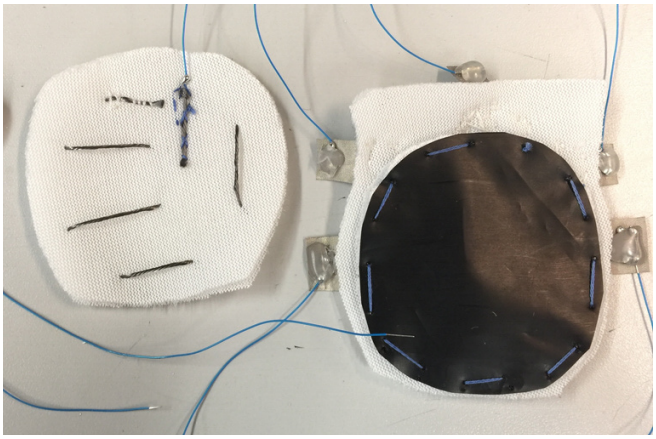
(Fg4. Conductive Fabric Sewed)



(Fg5. Wires soldered onto the conductive fabric)



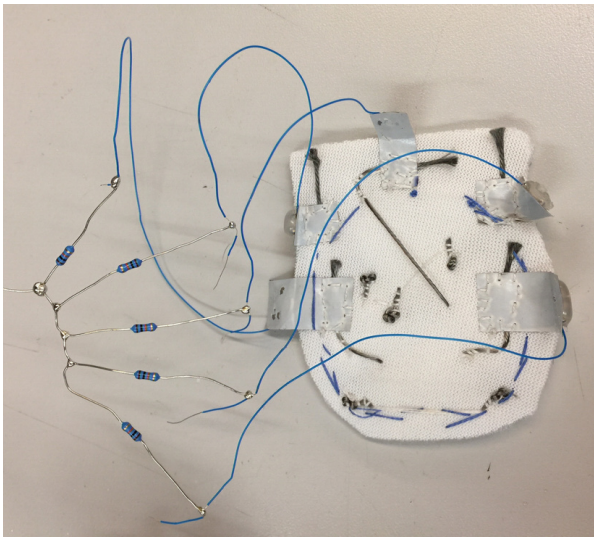
(Fg6. Hot glue applied)



(Fg7. Cleaned up)

After testing all the connections in this sensor is working, I cleaned them up, and I sewed everything together.

I soldered a 10K ohm resistor onto each wire in parallel connection to reduce the amount of current flowing to the microcontroller. By doing this, it reduces unwanted noise.



(Fg8. Resistors soldered)

I used the mozzi audio synthesis library to generate audio output according to the pressure value from the sensor. The example code and video of the working sensor can be found in the folder as this report.

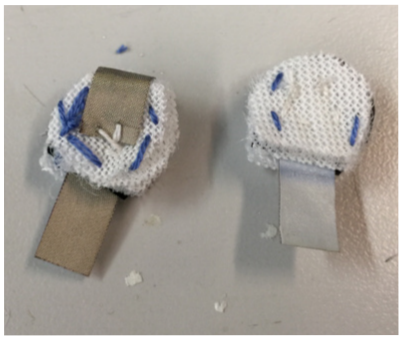
Fabric Button

Similarly to the pressure sensor matrix, the fabric button also allows me to build a tactile sensor using just the soft materials. Although this fabric button functions like a normal switch, the soft material works better with my overall design of the plushie. An LED will blink faster if users press its ears. It stimulates emotions like shyness. The interaction between users and this sensor brings characteristics to my plushie and makes it more humanlike.

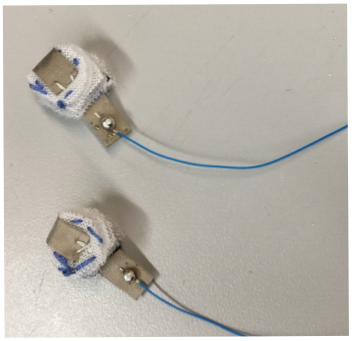
The logic behind the fabric button and the pressure sensor matrix are the same. However, the size of the ear is much small than the stomach, so instead of using the conductive thread. I directly used conductive fabrics to check the connection.

(Fg9. Materials prepared)

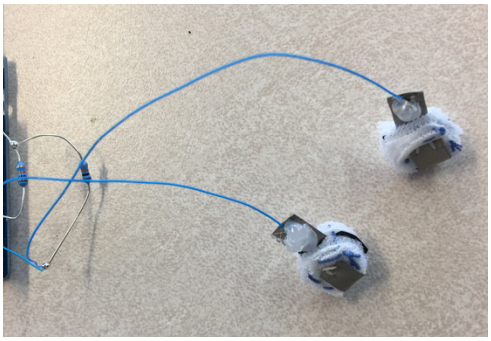
I assembled the layers with normal fabric threads and soldered a wire on each of them to connect to the Arduino. Similarly to what I had done for the pressure sensor, I soldered a resistor for each of them to reduce the noise.



(Fg10. Assembled)



(Fg11. Soldered)



(Fg12. Added resistors)

I have tested the connection and functionality of these fabric buttons with a multimeter and they are working well.

Review Proposal Idea and Prototype

When I started to build the prototype, I am converting my ideas into reality. It is different from what I planned at the proposal stage. Due to the lack of knowledge at the beginning of the prototype making stage, I spent too much time researching, exploring, and experimenting. I did not manage to accomplish all of the desired interactions I had from the proposal stage. My initial idea was to track users' frequency of inputting value into the musical interface and to regulate their mood by changing the color of plushie's body and the speed of the LED blinking on plushie's head. However, I realized that I am not able to do that due to lacking knowledge in Arduino and electronics, and the time constraint for this project. I refined my idea during the process of making this prototype. With I implemented fabric sensors into my prototype, it is now becoming a soft-material interactive musical plushie who have a more sustainable relationship with its users.

Future Improvements for Final Artifact

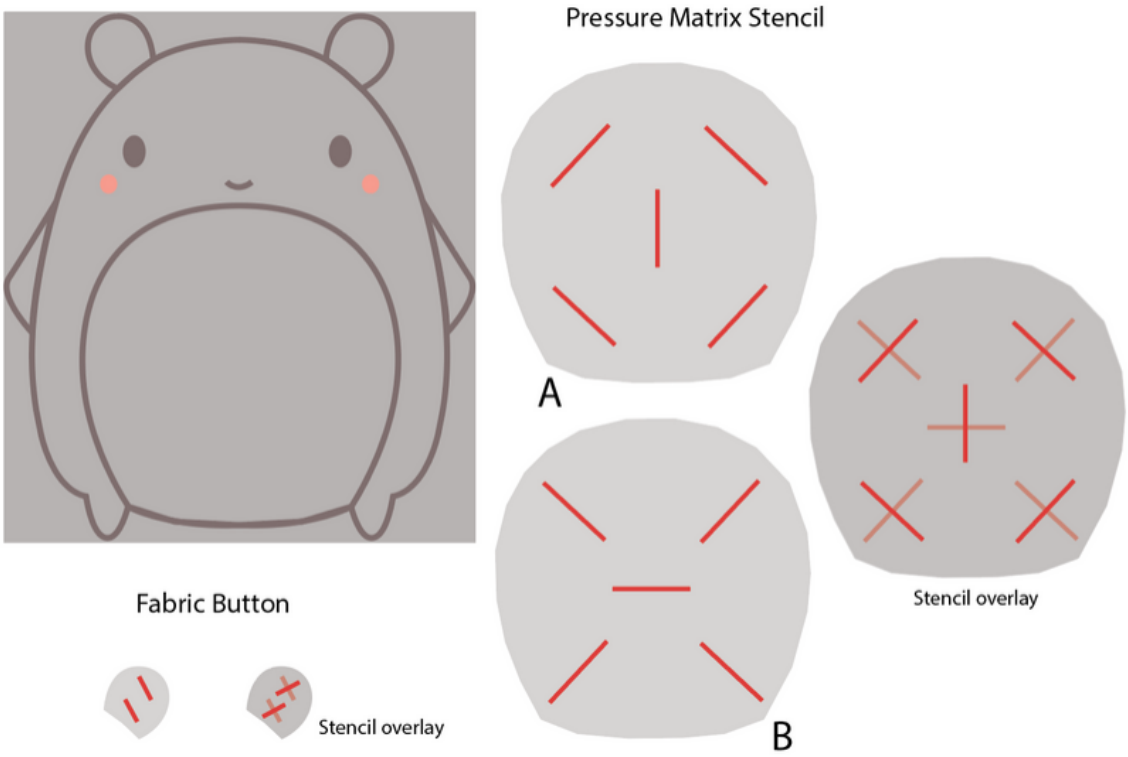
For the pressure sensor: The conductive threads do not cover the whole interface, there are areas on the interface do not give pressure value. I want to change the arrangement of the threads to improve the coverage.

For the mozzi audio synthesizer: I will have the working code for all the pressure input pins. And change them to make a more interesting experience for users. Moreover, I want to change the speaker to an embroidered fabric speaker to fit with the soft materiality of the plushie.

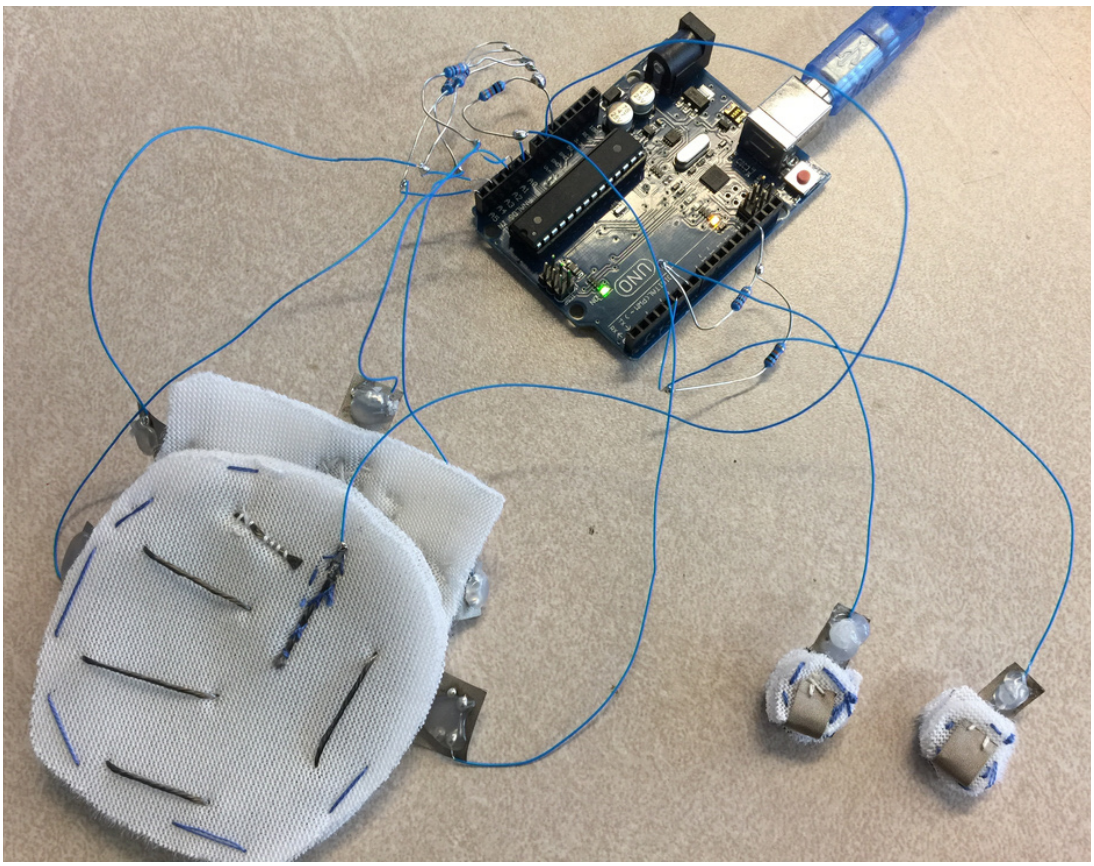
For the fabric button and the LED on the plushies' head: the LED blinks quickly when users squeeze the fabric button (plushie's ears).

For the eyes of the plushie: I will have the closed eyes sewed onto the plushies' body, and use thermochromic pigment for an opened eyes. When the temperature is high enough, the plushie looks like it opened its eyes.

Diagrams



(Fg13. Stencil Diagram)



(Fg14. Connections with the Arduino for prototype)

Reference

https://www.instructables.com/id/Pressure-Sensor-Matrix/
https://sensorium.github.io/Mozzi/
https://www.kobakant.at/DIY/?p=48