EMMA SHANNON

THE HEARTY PARTY

FABRICADEMY
textile and technology academy

ICELANDIC TEXTILE CENTER

CENTRINNO

Co-funded by the Creative Europe Programme of the European Union

TABLE OF CONTENTS

ICELANDIC TEXTILE CENTER

CENTRINNO

Co-funded by the Creative Europe Programme of the European Union

FABRICADEMY textile and technology academy

OUTPUT

THE HEARTY PARTY IS A WEARABLE SYNTHESIZER INSPIRED BY THE HEART

THIS PROJECT AIMS TO CREATE A DIALOGUE BETWEEN WEARABLE TECH, PARAMETRIC DESIGN AND TRADITIONAL CRAFT TO RAISE AWARENESS OF HEART DISEASE IN A FUN AND PLAYFUL WAY

THE HEARTY PARTY IS THE FINAL PROJECT FROM EMMA SHANNON FOR FABRICADEMY 2022/2023

ICELANDIC TEXTILE CENTER

CENTRINNO

Co-funded by the Creative Europe Programme of the European Union

FABRICADEMY textile and technology academy

INTRODUCTION

EMMA SHANNON

Originating in Leith in Edinburgh SCO.
After living in several British cities I moved to Reykjavik in Iceland 10 years ago.
After dropping out of 2 biology 1st years early on in life and having no training in the arts, I found myself achieving a BA in Weaving and Textile Design from Glasgow School of Art.

Passionate about contemporary music and culture, I have been involved in many projects in upcycling, art , event management and fashion.
I have been working for 28 years in the beer industry and was at the forefront of the craft beer revolution in Iceland.

In June 2022 I had a sudden heart attack. This has obviously been a profound and life changing experience that has sent me on this journey through Fabricademy.

I have been wanting to do this course since taking part in the bootcamp here in 2019. Due to my recovery I actually had the time to leave the beer industry and explore these new parameters in my life.

Rather than my usual artistic influences, I decided to own my new situation and use the heart and cardiac issues as my inspiration and inputs, to educate myself and others about heart disease but also as another therapeutic tool to understand my new reality.

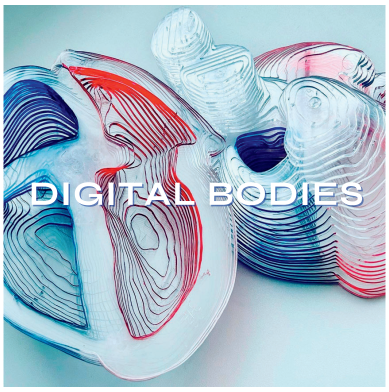
Hope you enjoy The Hearty Party as much as I did.

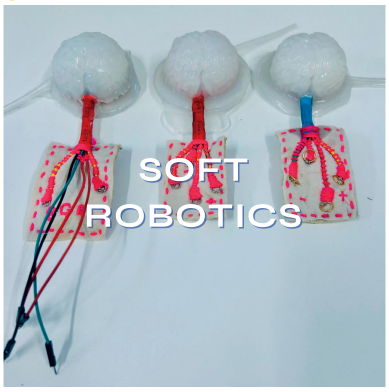
ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

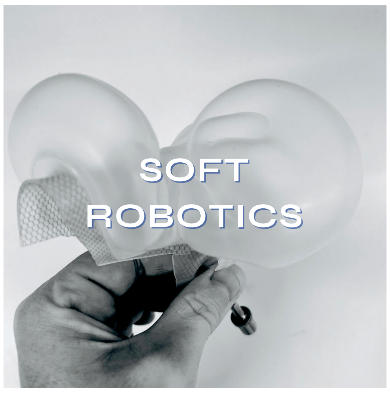
THE FABRICADEMY JOURNEY

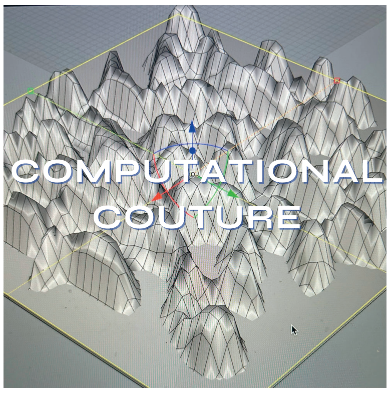
Through the initial weeks of the Fabricademy course I created a series of projects keeping the heart as the focus. I learned so many new skills during this process but I was especially drawn to E-Textiles, 3D Printing, Mould Design and the manipulation of Fish Leather, a local resource. These are the areas I decided to focus on for my final project design and research.

Here are some of my favourite outcomes from the work that I created during our 13 weeks of assignments.









ICELANDIC TEXTILE CENTER
(1) (2) CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

WHAT ABOUT THE HEART?

Heart disease is the largest cause of death, not only in Iceland but also world wide. (1) (2)
Women often display different symptoms to men and are up to 50% more likely to be misdiagnosed. (3) (4)
I myself did not display the classic symptoms normally associated with a heart attack. I thought it was just very bad indigestion as I had not been made aware of any alternative explanations of the experience. I was also misdiagnosed for 6 days with a stomach ulcer.
While researching for Fabricademy I found that out this is more common an issue than you would of thought.

This is why I wanted to use my experiences and the new found skills I learned at Fabricademy for awareness, creating an interesting way of making all people and especially women like me think about their heart.

After experiencing heart disease many of us can feel ashamed or judged, as it is considered as an indicator of an unhealthy lifestyle. You don´t talk about your heart health and people don´t want to talk about your heart health either. You also spend alot of time in the early days of recovery being monitored with sticky pads on, doing stress tests on cardiac bikes and generally learning new ways of living with the condition. Thats why I was determined to make this project as fun as humanly possible. My new mantra was "not dead yet" and I wanted to reflect this attitude in the work.

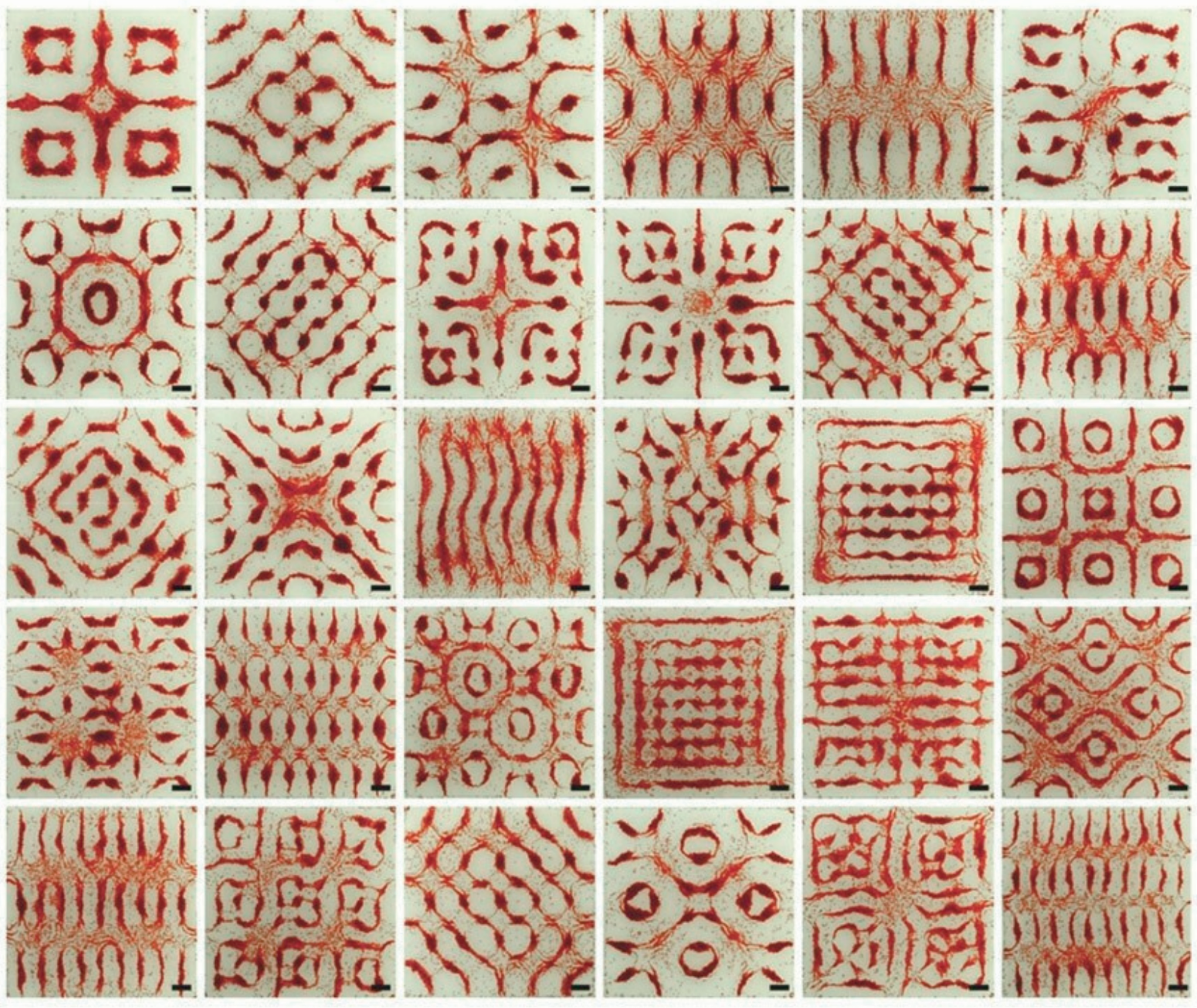
It is my greatest wish that this project will highlight and inspire a dialogue surrounding heart disease in women and its issues. I am especially delighted to have brought a little fun and joy to those affected by this issue as I have been.

(1) Statistics Iceland, 25/10/21, Diseases of the Circulatory System and Neoplasms the Main Causes of Death in the Past 10 Years.
(2) World Health Organisation, Cardiovascular diseases.
(3) British Heart Foundation, Misdiagnosis of Heart Attacks in Women.
(4) O'Connor, Anahad. The New York Times, 9/6/22, Why Heart Disease in Women Is So Often Missed or Dismissed.

ICELANDIC TEXTILE CENTER CENTRINNO Co-funded by the Creative Europe Programme of the European Union FABRICADEMY textile and technology academy

HEARTSPIRATION

One of my main inspirations for all of my explorations during the course is a study at Stamford (5) showing acoustic choreography in heart cells. A simple change in frequency and amplitude manipulates heart cells into intricate patterns exploiting a type of acoustic signal that creates Faraday waves that result from a physical perturbation at the interface of liquid and air. This study inspired me into looking at ways that the heart reacts with sound and also how this data can be used in parametric design to form shape and pattern.

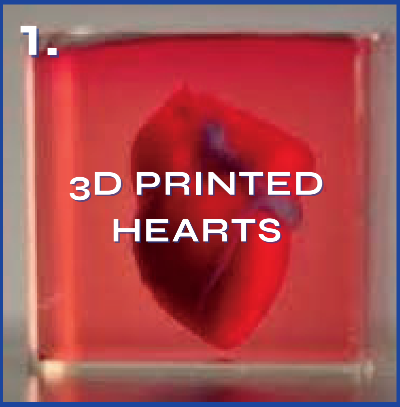


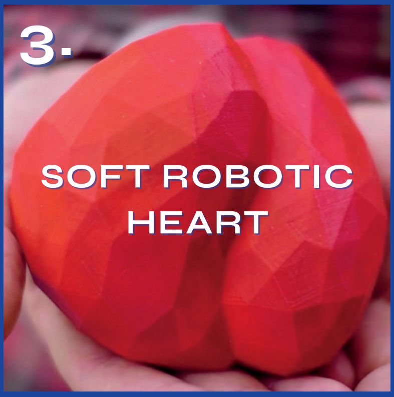
(5) Armitage, Hanae, 21/6/18 Stamford Medical Magazine. Sound research Scientific Innovations Harness Noise and Acoustics for Healing. Wu, Sean, MD, PhD, and Demirci, Utkon, PHD. Acoustic Choreography.

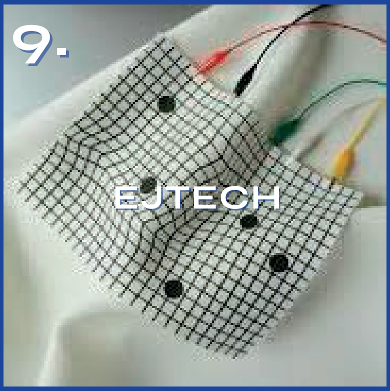
ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

HEARTSPIRATION

OTHER RESEARCH THAT INFORMED THIS PROJECT



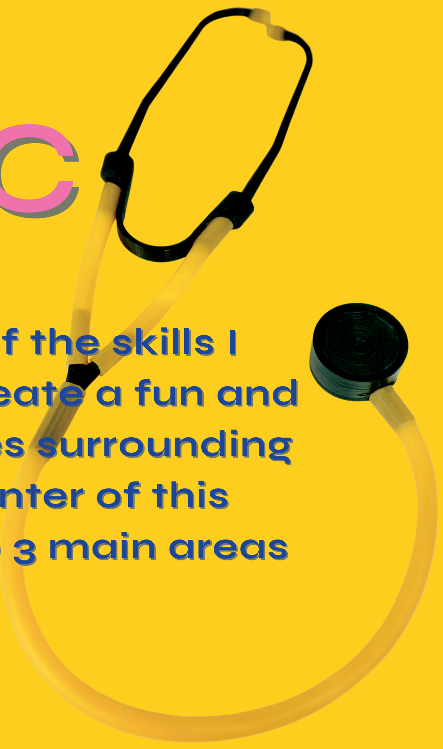




1. Dvir, Tal. Shapira, Asaff. 15/4/19. Advanced Science. 3D Printing of Personalised Thick and Perfusable Cardiac Patches and Hearts. (6)
2. Xudong, Wang. Jung, Li. 21/7/21. Advanced Functional Materials. Multifunctional Artificial Artery from Direct 3D Printing with Built-In Ferroelectricity and Tissue-Matching Modulus for Real-Time Sensing and Occlusion Monitoring. (7)
3. King's College London, Rusty Squid, the Royal Academy of Engineers and the British Heart Foundation. 14/2/18. Heart in Your Hands Project. (8)
4. Björk. Merry, James T. Huang, Andrew. 19/06/18. Still from Family/Black Lake Video. Vulnicura. One Little Indian Records. (9)
5.Ýrúrarí. Heimisdóttir, Ásgerður. 2021. Facing Winter. (10)
6. Maclachlan, Lynne. 2021. Quiver Necklace, Pink. (11)
7. Stanley, Jessica. 2019. Stitchsynth. Fabricademy (12)
8. Cat Full of Ghosts Electronics. 23/6/20. Yowler DIY Noise Synth (13)
9. EJTech. 2017. Liquid Midi. Sound and Matter in Design Museum, Holon. (14)

ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

PROBLEMATIC



For my final project I wanted to bring some of the skills I had learned during the course together to create a fun and engaging piece to highlight some of the issues surrounding heart disease. Using the heart itself at the center of this exploration, I decided to split this inquiry into 3 main areas of research.

FEEL THIS?

The heart forms the basis of experimentation in computational parametric design. Anatomical information from cardiac forms and functions is used to create new shapes and 3D printed surfaces to be moulded in fish leather. This skin on skin interaction brings the heart out onto the chest as a tactile and interactive display.

SEE THIS?

These fish leather moulded pieces serves as a visual representation of the heart outside the body. The connections made between these new forms represent the arteries and blood flow. The heart is also visualised in the display of light linked to the sonic parameters.

HEAR THIS?

Inspired by my own experiences of being monitored in various ways for my own heart disease, auscultation sounds of the heart are isolated and used in an interactive way forming another dialogue between the inner and outer world. Finally, asking the question, If the heart could speak what would it say?

`

ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

WHY?

FISH LEATHER

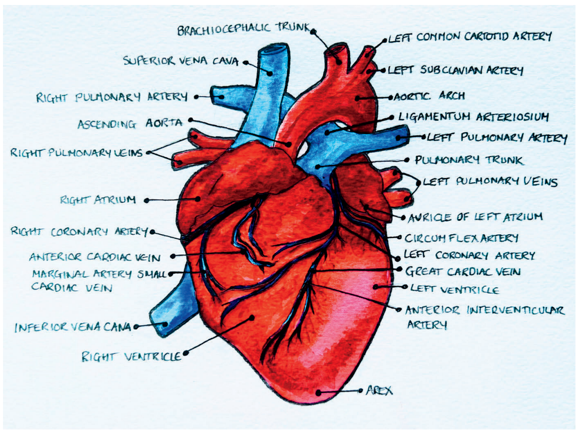
Fish Leather is an incredible and sustainable local resource in Iceland. It is a product that is made from discarded fish skins from the fishing and fish farming industries. I bought mine from the producer, Nordic Fish Leather (15) in Sáuðárkrókur, a town near Blönduós where we have been living while on the Fabricademy Journey. Through my research, I believe that this resource is still underexplored as a material. It has been used mainly for bags and shoes by luxury brands and locally for some jewellery pieces aimed at the tourist market.
During the making of The Hearty Party, I have really enjoyed learning the properties of fish leather through moulding and sculpting 3 dimensional pieces. .
I also used local Icelandic wool in this project to make it as environmentally aware as is possible.

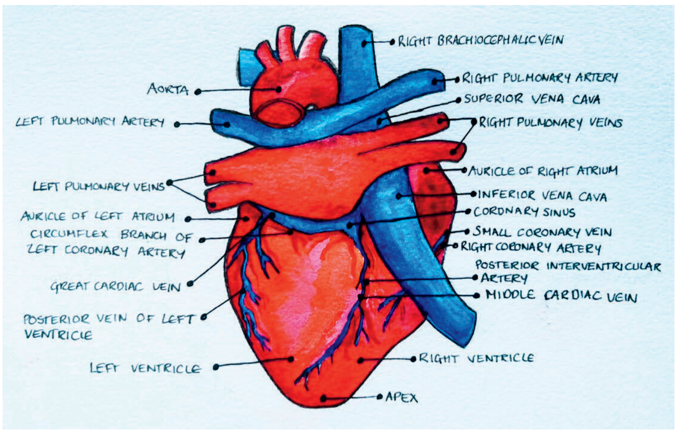
Iceland has been my home for the last 10 years and I am looking forward to future experimentations and collaborations with these materials.

ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

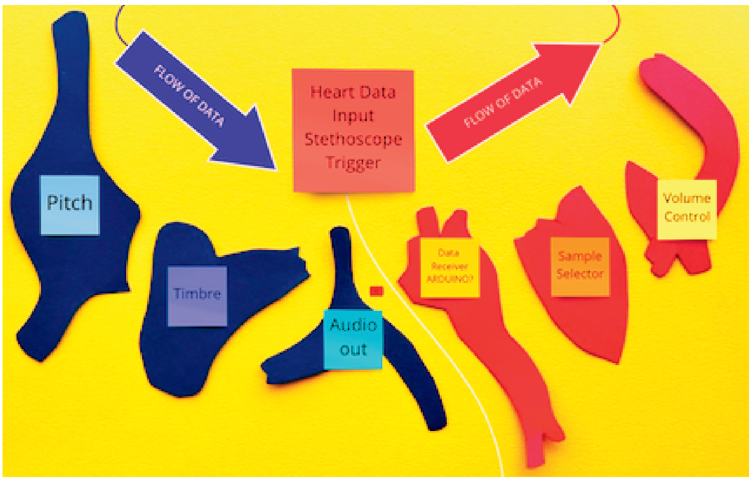
FEEL THIS

The first thing I did to get the form of the project was take myself back to anatomy school. I studied Biomedical Science many years ago and I was always fascinated by the human body. Understanding the heart and and the way it works was fundamental in the design from the shape and also the way it behaves.
My first step was to make some drawings of the heart.





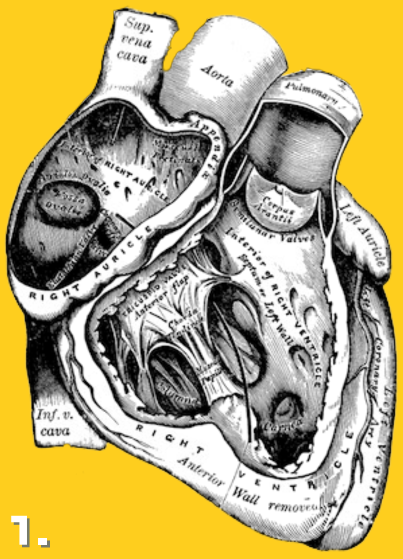
My next step was to separate the parts to find ways to achieve the separate chambers I wanted to create in the piece. I also started to think of possible ways to embed and use sound and how that would work in a way that would follow the heart functions as shown in this initial schematic.



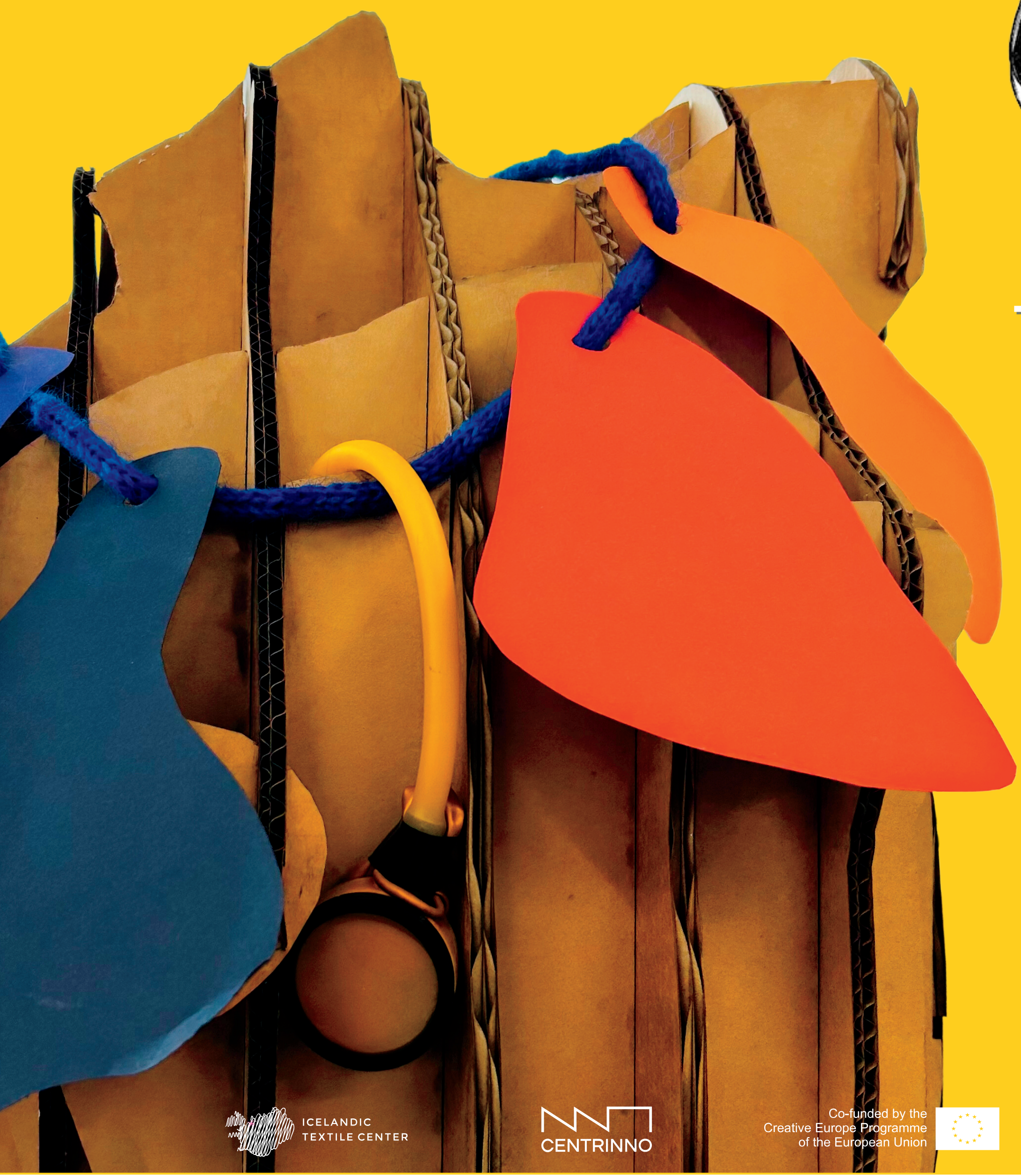
WEARABLE

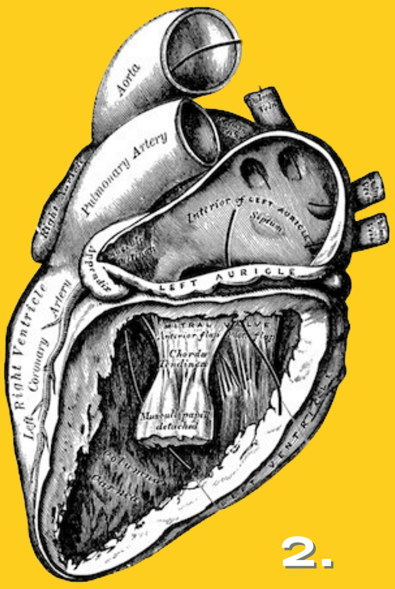
Using these anatomical references I started to play with the placement on the body. I wanted the piece not to be too obviously heart shaped and to sit nicely on the chest. At this point I went back again to my origins and to the illustrations from the anatomy bible, Greys Anatomy (16). Using these drawings I used the shape of each ventricle and atrium to get a basic shape that I rotated to get a nice potential wearable neckpiece.

1. Gray, Henry. 1913 Gray's Anatomy: Descriptive and Applied. Philadelphia: Lea & Febiger. pg.555.
2. Gray, Henry. 1913. Gray's Anatomy: Descriptive and Applied. Philadelphia: Lea & Febiger. pg.560.



1.

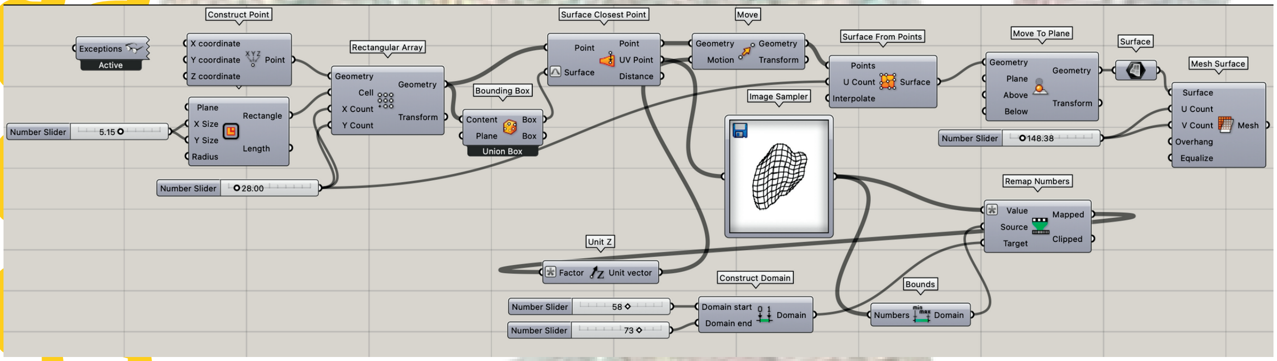




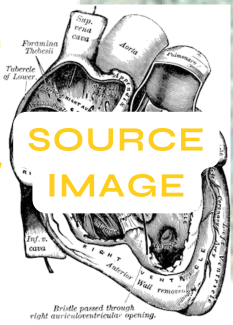
2.

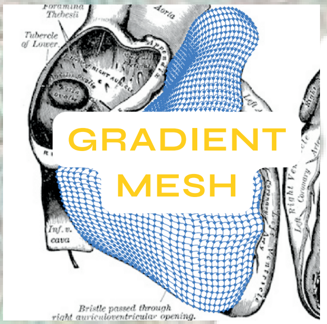
PARAMETRIC

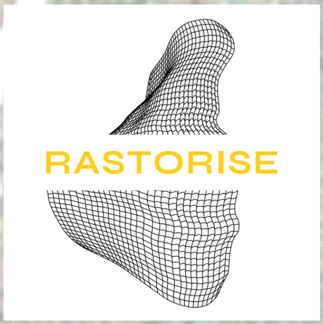
I played with many definitions in Rhino and Grasshopper to achieve the look I wanted for my forms. I went back to a definition I had developed earlier in the course to create CNC moulds using a component called Image Sampler to use a 2D image to create 3D forms. This component works by assigning values to different colour paths, in this case black and white. In my definition the white, in this example stays at zero and the black areas are lifted up in the Z axis.

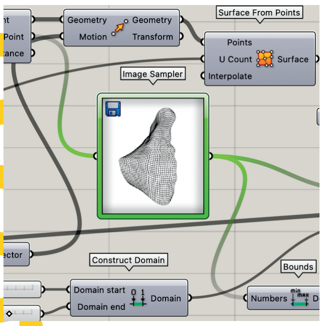


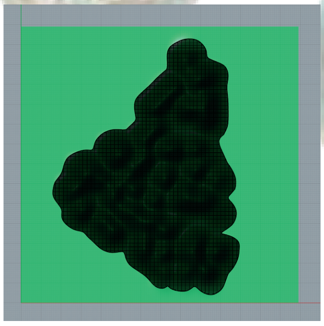
After much experimentation I came up with a system that worked for me. Firstly, by using Illustrator to get the image and then placing this in the Image Sampler. I then extruded this in Rhino.

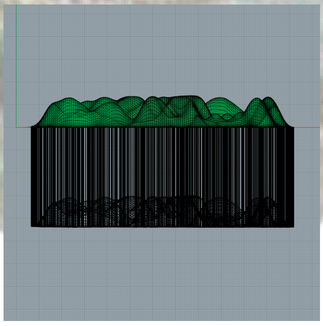




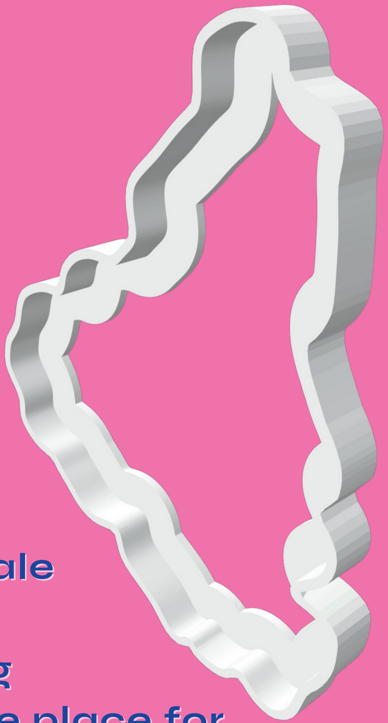






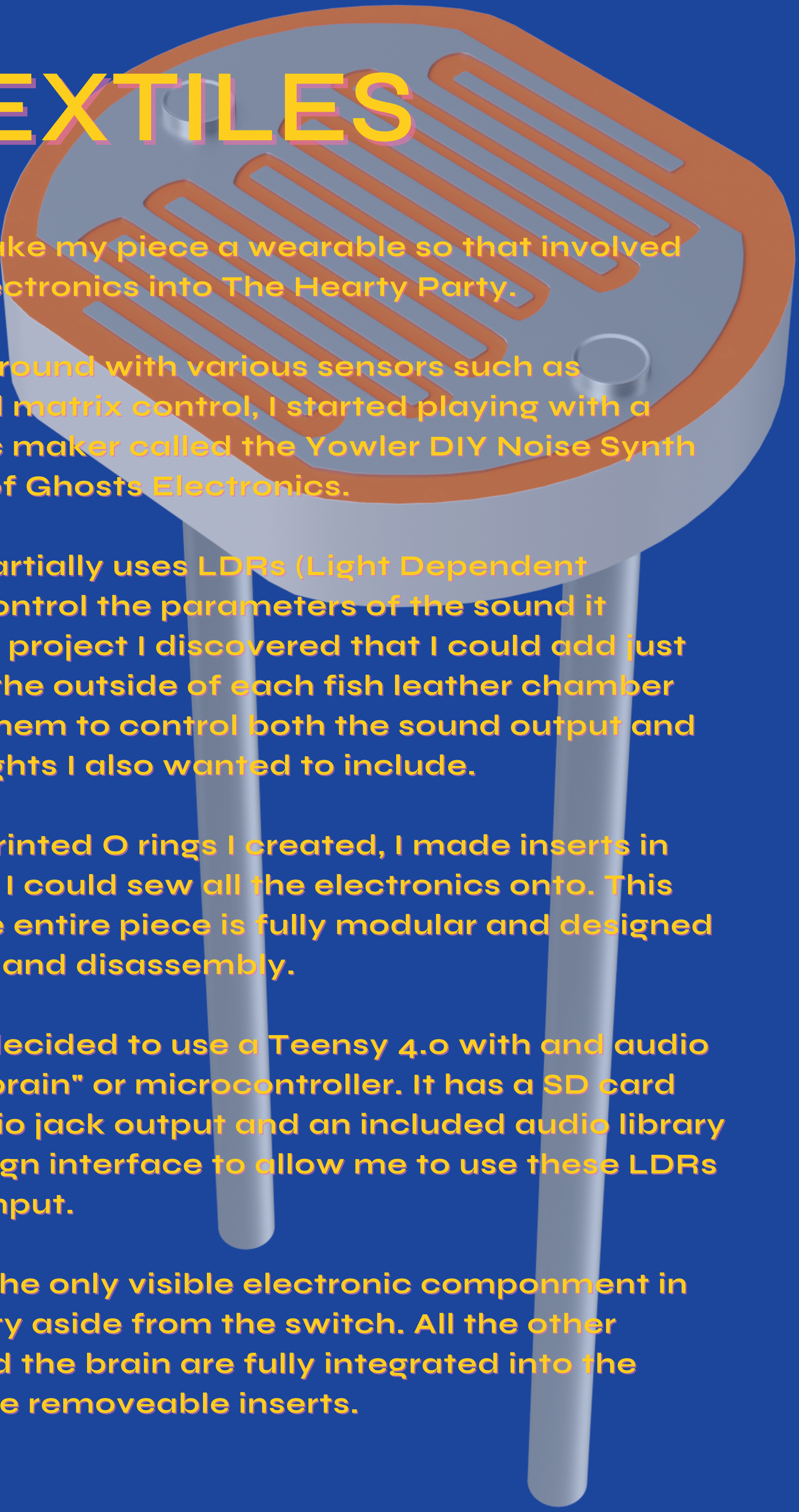


3D PRINTING



These 3D models are 3D printed on a Original Prusa i3.
Using 3D printing technology allows me to prototype locally in Iceland where small scale manufacturing is essential.
I also created a series of companion O-Ring pieces in Grasshopper that enabled a stable place for embedding electronics and another mould for the base of these fish leather chambers.
I printed these pieces with a 10% grid infill as there is no need for them to be very sturdy for this purpose. The print was relatively quick and light in use of resources. All my moulds were printed using Add North Premium Silk Filament in red. I have already reused these moulds many times.

E- TEXTILES



I wanted to make my piece a wearable so that involved embedding electronics into The Hearty Party.

After playing around with various sensors such as capacitive and matrix control, I started playing with a modular music maker called the Yowler DIY Noise Synth from Cat Full of Ghosts Electronics.

This module partially uses LDRs (Light Dependent Resistors) to control the parameters of the sound it outputs. In this project I discovered that I could add just one LDR onto the outside of each fish leather chamber and program them to control both the sound output and the neopixel lights I also wanted to include.

Using the 3D printed O rings I created, I made inserts in Neoprene that I could sew all the electronics onto. This meant that the entire piece is fully modular and designed for both repair and disassembly.

At this point I decided to use a Teensy 4.0 with and audio shield as the "brain" or microcontroller. It has a SD card reader, an audio jack output and an included audio library and audio design interface to allow me to use these LDRs as my sensor input.

The LDRs are the only visible electronic componment in the hearty party aside from the switch. All the other electronics and the brain are fully integrated into the chamber on the removeable inserts.

.

ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

SEE THIS

Velleman Bright Dot Neopixels lights were embedded into the chambers to provide a form visual communication and also as an aid with calibration. Also, lights are cool.

I found it very difficult to program these particular products as there is very little useful documentation online.

I used 4 neopixel strips. 2 with 3 lights and 2 with 5 lights. Each strip was assigned into a particular chamber. the 2 smaller chambers got the strips with 3 lights in them.

Through trial and many errors I was able to make each one of these lights fully addressable meaning I could made each individual light in each strip light up in a particular colour depending on the values it received.

The lights were programmed to be responsive to the audio mode they were in (more about these audio modes later in the booklet).

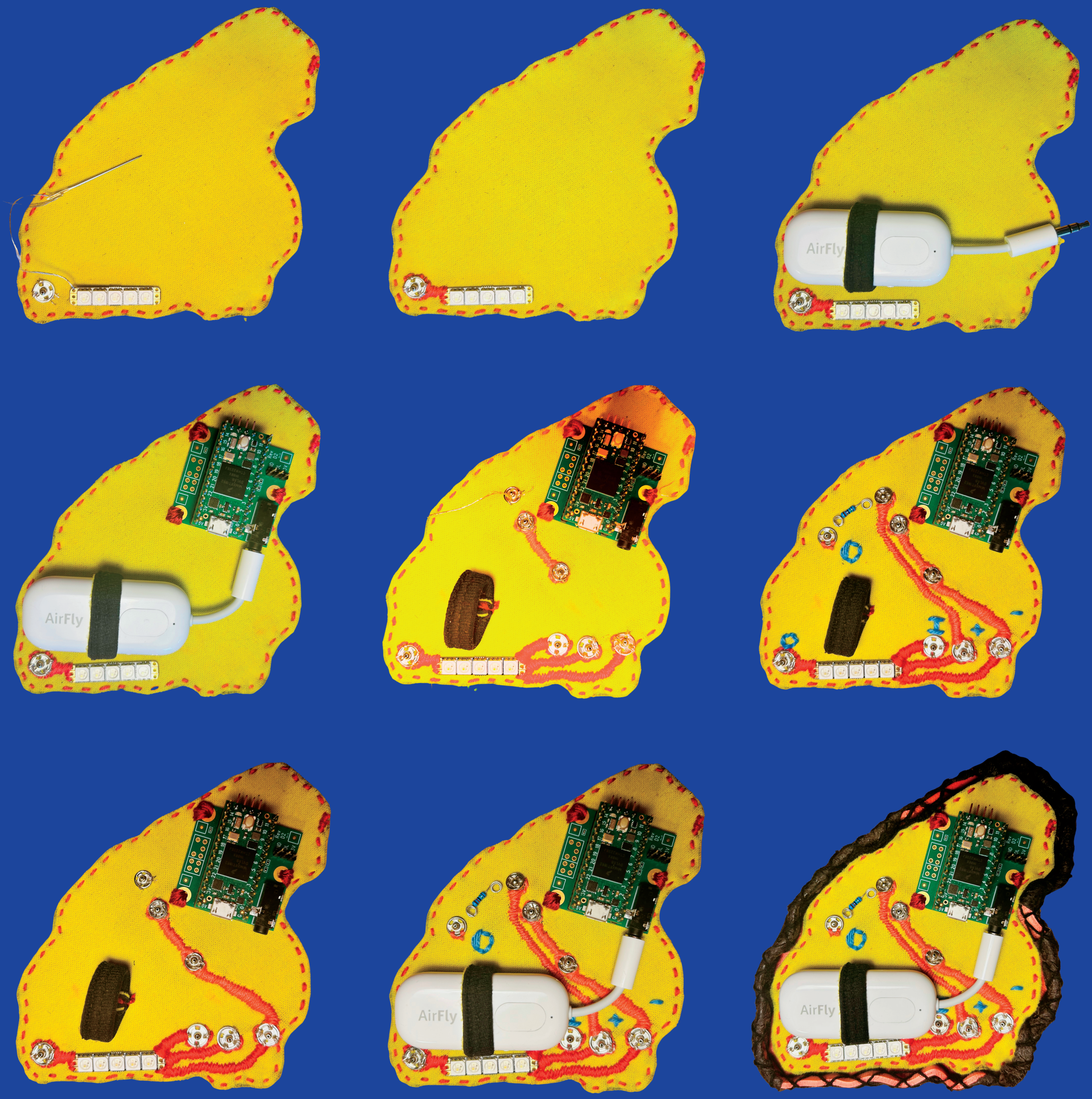
I was able to get them to respond to values so as you pass your hand over the LDR and therefore changing the resistance you can change the colour, make them flash and have them on full on party mode.

It actually became an important consideration later to be able to calibrate the LDRs so they would function in different light levels. Having the neopixels embedded allowed for a visual calibration tool where, in this instance, once switched on, the lights turn white. At this point you caver all the LDRs. Once the rainbow cycle starts you remove the cover and they are calibrated to the ambient light you are in. If you move you need to recalibrate.

.

ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

DONT SEE THIS



I hand-sewed all the circuits into the inserts and then covered them with satin stitch making sure there was little opportunity for short circuits.
All of the connections were sewn in with poppers allowing for the easy connection and disassembly.

ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

HEAR THIS

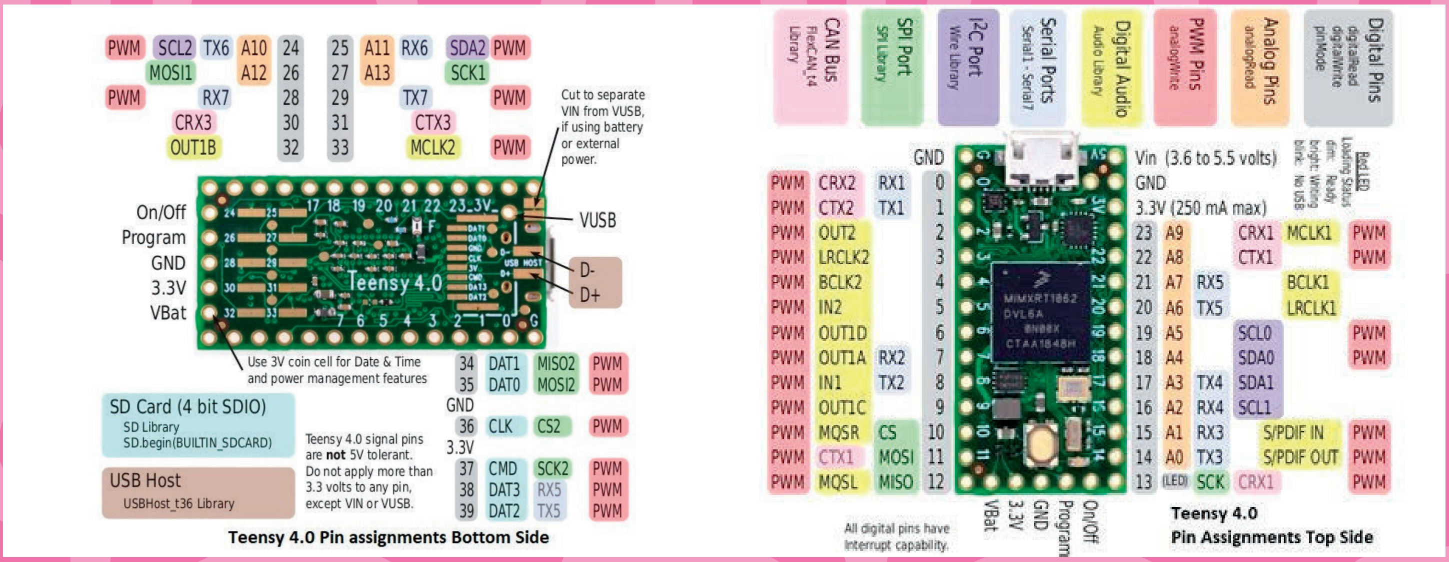
As a heart patient you are constantly monitored for the sounds of you heart as auscultation is essential in diagnosis of cardiac issues. My main motivation in this inquiry at the beginning was to listen to the heart but move away from the sticky wound inducing pads we all endure.

This is was what gave me the idea to play with the sound of my heart to explore what it had to say. I tried to isolate my own heart beat as a trigger to set off a array of samples but I found that it is incredibly difficult to isolate the heart even with the stethoscope I made myself using the Stethoscope from the Open Source Gija project (6). This is an amazing project that I utilized and made a great contact mic from but unfortunately it didn´t give me values that I could use during tests with users.

I actually got the sounds of my own echocardiogram from my cardiologist but unfortunately these were tied to programs only available at a hospital so instead I downloaded the sounds of myocardial infarction (heart attacks) from Medscool (7) on youtube.

I assigned these sounds to each chamber but I didn´t think this was "fun" enough. I decided to add a sample of Elton Johns Don´t Go Breaking My Heart as an easter egg so that if you cover all the LDRs at once, the piece enters Elton Mode, allowing you to change the dynamics of the music sample. I isolated a tiny bit of the song and uploaded it onto the Teensy via the SD card reader on the inbuilt audio shield.

BRAINS?



I prototyped on a Teensy 4.1 but learning all the capabilities of this programming development board in its various incarnations was loads of fun.

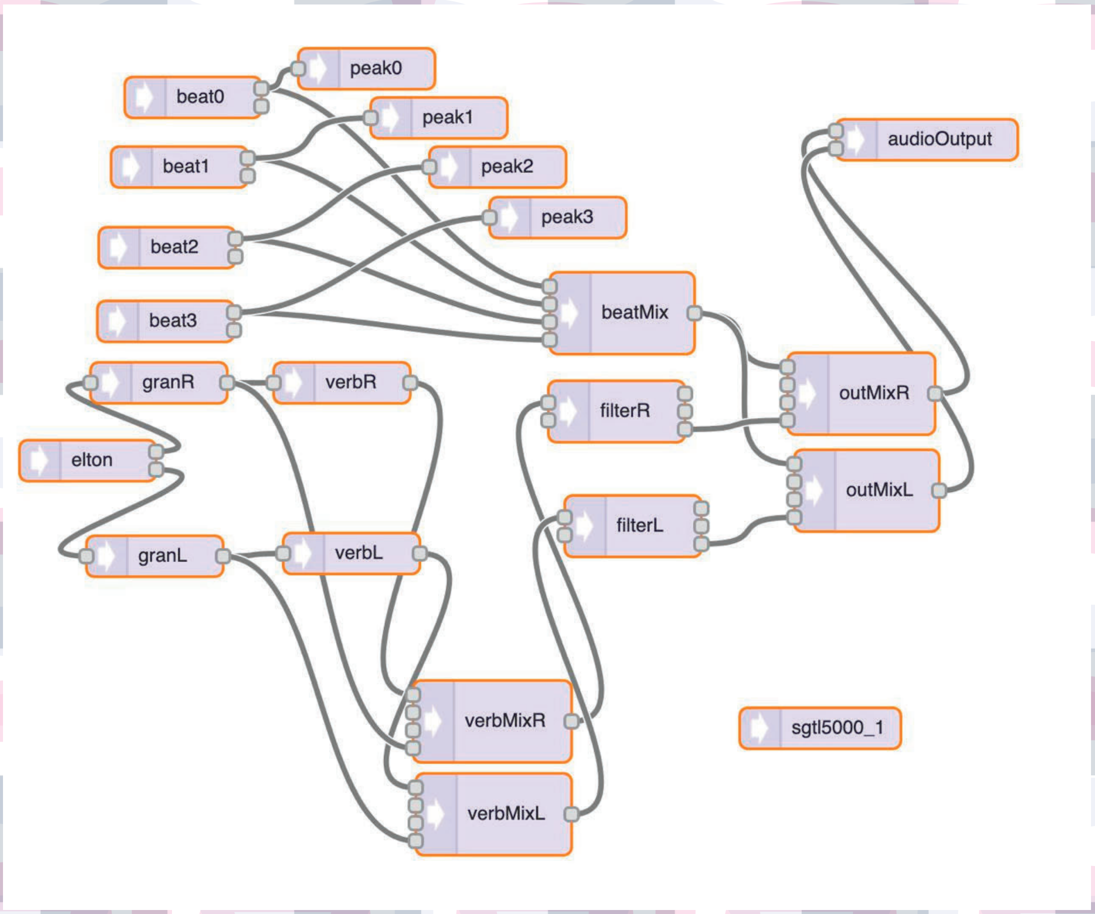
Figuring out the pins and what they do was part of the process. you have to be mindful of analog and digital capacity so its best to figure out what is the right development board for your needs.

The 4.1 is actually much larger than the 4.0, with way more inputs. this turned out to be unnecessary in this project. The LDRs needed 4 data input pins and the the LEDs required 1 so there was more than enough on the smaller 4.0 Teensy microcontroller.

Finding the ways to embed this microcontoller on the inserts parts I made while figuring out how it all gets power was an education was an absolute journey in soft circuitry. I´ve learned so much in how to do and not to do e-textles from this.

ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

GUI



The Gui interface allowed me, a non musician, to fully understand visually how the sound would be broken up and gave me a visual picture of how the audio would work. The best thing is that once you have created the path for your audio here you can upload the code directly into Teensyduino .

Teensyduino is a offshoot of regular Arduino IDE and its fully compatible. You just need to download the teensy uploader and have this running at the same time as the teensyduino.

Sometimes all this can seem overwhelming. it certainly was to me so I hope you will look at my documentation online at the download code at the rear of this booklet.

ELTON MODE

Having heart beat samples installed and working well I really felt like it needed to be more fun. Thats when I got the idea of using a sample from Elton John and Kiki Dees 1976 duet, Don´t Go Breaking my Heart. Written by Elton John and Bernie Taupin under the pseudonyms "Ann Orson" and "Carte Blanche". I actually saw Elton John´s Goodbye Yellow Brick Road tour 2 days before my Heart Attack so for me it was the perfect sample to use. It is conceived as an "easter egg" so when all of the LDRs ar covered it enters "Elton Mode". I also programmed it so that if you are not interacting with it, it returns back to the heart beat samples. When in Elton Mode without interaction the Neopixels are cycling through a bright rainbow array of colours.

Isolating a small sample of the song I uploaded it onto the SD card and then decided on 4 different audio parameters to alter for the 4 different chambers.

One chamber is programmed as a reverb effect. When this is in action the neopixels slowly morph between colours.

Another has a high pass filter that when activated the colours gradually desaturate.

The next two chambers actually work together using granular synthesis. One as a granular freeze that make the neopixels flash between blue and red. The other is granular pitch shift that changes the frequency of this flash.

THE CODE

```
 // Libraries
#include <Adafruit_NeoPixel.h>

// LED stuff
#define NUMPIXELS 16 // <------ CHANGE ME
#define LED_PIN 26
#define brightnessValue 150
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, LED_PIN, NEO_GRB + NEO_KHZ800); // This code is
always included when working with neopixels

//const int pixelGroups[5] = {0, 0, 1, 2, 3}; // Assign each LED to a group.
//const int pixelGroups[8] = {0, 0, 0, 1, 1, 2, 2, 3}; // Assign each LED to a group.
//const int pixelGroups[11] = {0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 3}; // Assign each LED to a group.
//const int pixelGroups[16] = {0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3}; // Assign each LED to a group. 3-5-3-5
const int pixelGroups[16] = {0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3}; // Assign each LED to a group. 3-5-5-3

// The pixel colours in ELTON mode are stored here. This is needed to mix colours from different effect modes.
unsigned int COLORS[NUMPIXELS][3] = {};

// DEBUGGING CODE
const bool DEBUG = true;
const bool DEBUG_ELTON = false;

////////////
// Setup //
////////////
void setup() {
 Serial.begin(9600);

 // Lights Development
 pixels.begin(); // This initializes the NeoPixel library.
 pixels.setBrightness(brightnessValue); // full brightness=255 (not recommended as its very very bright)
 pixels.show(); // Initialize all pixels to 'off'

 int LED_STARTUP_TIME = 50;

// LDR Calibration Start up Mode
 Serial.println("Turn all LEDs to WHITE");
 colorWipe(pixels.Color(255, 255, 255), LED_STARTUP_TIME * 5); // White
 delay(3*LED_STARTUP_TIME);
 Serial.println("Turn all LEDs off");
 colorWipe(pixels.Color(0, 0, 0), LED_STARTUP_TIME); // Off
 delay(10*LED_STARTUP_TIME);

 int LDR_1[] = {1023, 1023, 1023, 1023};
 rawLDRValues(LDR_1);

// Pixel introduction.
 Serial.println("Turn all LEDs to RED");
 colorWipe(pixels.Color(255, 0, 0), LED_STARTUP_TIME); // Red
 delay(3*LED_STARTUP_TIME);
 Serial.println("Turn all LEDs to GREEN");
 colorWipe(pixels.Color(0, 255, 0), LED_STARTUP_TIME); // Green
 delay(3*LED_STARTUP_TIME);
 Serial.println("Turn all LEDs to BLUE");
 colorWipe(pixels.Color(0, 0, 255), LED_STARTUP_TIME); // Blue
 delay(3*LED_STARTUP_TIME);
 Serial.println("Turn all LEDs to WHITE");
 colorWipe(pixels.Color(255, 255, 255), LED_STARTUP_TIME); // White
 delay(3*LED_STARTUP_TIME);
 Serial.println("Turn all LEDs off");
 colorWipe(pixels.Color(0, 0, 0), LED_STARTUP_TIME); // Off

 // Teensy does not recommend using this, since the increased resolution will mostly be noise.
 // https://forum.pjrc.com/threads/25111-Decreasing-noise-on-Teensy-ADC?p=42891&viewfull=1#post42891
 //analogReadRes(16); // Teensy 3.0: set ADC resolution to this many bits

 // NYI: This might be useful to keep the input signals a bit more "steady".
 //analogReadAveraging(16); // average this many readings

 // Audio connections require memory to work. For more
 // detailed information, see the MemoryAndCpuUsage example
 AudioMemory(AUDIO_MEMORY);

 // Enable and set the volume of the Audio shield output.
 sgtl5000_1.enable();
 sgtl5000_1.volume(OUTPUT_VOLUME);

 // The Granular effect requires memory to operate. Must be pre-allocated.
 granR.begin(granMemR, GRANULAR_MEMORY_SIZE);
 granL.begin(granMemL, GRANULAR_MEMORY_SIZE);
```

```
// Set the High Pass filters.
filterR.resonance(FILTER_RESONANCE);
filterL.resonance(FILTER_RESONANCE);
filterR.frequency(0);
filterL.frequency(0);

// Set initial volumes for the dynamic volume mixer, BeatMix.
muteBeatMix(); // Set the initial beatMix levels to 0.
setVerbDW(0.0); // Set Reverb fully DRY

// Set static mix volumes. These should not need to be changed.
setOutMixChannel(0, 1.0); // Forward the beatMix audio at unity volume.
setOutMixChannel(1, 0.0); // Unused channel set to 0.
setOutMixChannel(2, 0.0); // Unused channel set to 0.
setOutMixChannel(3, 1.0); // Forward the ELTON mode audio at unity volume.
verbMixR.gain(1, 0.0); // verbMix channel not used.
verbMixR.gain(2, 0.0); // verbMix channel not used.
verbMixL.gain(1, 0.0); // verbMix channel not used.
verbMixL.gain(2, 0.0); // verbMix channel not used.

// Get the ambient LDR readings for Calibration.
rawLDRValues(LDR_0);
// Add a small value to the initial LDR readings in order to emphasise the zero value at ambient conditions.
for(int k = 0; k < 4; k++){
LDR_0[k] += AMBIENT_PADDING;
LDR_M[k] = LDR_1[k] - LDR_0[k]; // Populate the LDR_M (LDR_Max) array for calibration.

// If the difference is small, the calibration has failed. Setting to a more wide value.
if(LDR_M[k] < AMBIENT_PADDING){
LDR_M[k] = 1023 - LDR_0[k];
}
}

// Some SD card testing. Directly taken from the example.
SPI.setMOSI(SDCARD_MOSI_PIN);
SPI.setSCK(SDCARD_SCK_PIN);
if (!(SD.begin(SDCARD_CS_PIN))) {
// stop here, but print a message repetitively.
while (true) {
Serial.println("Unable to access the SD card");
delay(500);
}
}

// Stop if the samples on the SD card can't be read.
if(!SD.exists(BEAT0_SAMPLE) or !SD.exists(BEAT1_SAMPLE) or !SD.exists(BEAT2_SAMPLE) or
!SD.exists(BEAT3_SAMPLE) or !SD.exists(ELTON_SAMPLE)) {
while (true) {
Serial.println("Unable to load file(s) from SD card.");
delay(500);
}
}
}


/////////////////
// Main Loop //
/////////////////
void loop() {

// Start the loop by getting the current LDR ratios from calibration.
storeLDRRatios();

//delay(100);

// Set the mix, depending on the LDR values.
setBeatMix();

// Loop all the sounds. This is rough way of doing it.
// Use a different library to be able to adjust the playback speed.
if (!beat0.isPlaying()) {
beat0.play(BEAT0_SAMPLE);
}
if (!beat1.isPlaying()) {
beat1.play(BEAT1_SAMPLE);
}
if (!beat2.isPlaying()) {
beat2.play(BEAT2_SAMPLE);
}
if (!beat3.isPlaying()) {
beat3.play(BEAT3_SAMPLE);
}
```

```
// Every PEAK_TIME milli seconds, update the status of the peak detecting LEDs.
// Can be done more easily with milliseconds.
if (msecs % PEAK_TIME == 0) {
blinkHeartBeatLEDs();
}

if (triggerElton()){
Serial.println("Trigger Elton Mode!");

// Stop and mute all the beats playing. We're in ELTON MODE now!
beat0.stop();
beat1.stop();
beat2.stop();
beat3.stop();
muteBeatMix();

// Initialise LED variables.
unsigned int wheel_pos = 0; // Needed for the rainbow effect. "Counts" the position on the color wheel.
unsigned int last_rainbow_tick = millis() - RAINBOW_INTERVAL;
unsigned int blink_time = millis(); // Needed to blink stuff.
unsigned int blink_interval = 150;
bool blink_state = true;
uint8_t slow_wheel_pos = 0;
unsigned int slow_wheel_tick = millis();

// Effects variables
bool freeze_enabled = false;

// Grace period
unsigned int grace_period = millis() + ELTON_GRACE_PERIOD_TIME;

// The isPlaying function of the sample player sometimes takes a few milliseconds to represent the correct value.
// Usually this is solved with a delay in example codes. But delays don't work well here.
unsigned int is_playing_compensation = millis() + 100;

// Loop while the ELTON sample plays.
while(remainElton() or elton.isPlaying() or millis() < is_playing_compensation){

// Loop the ELTON sample
if (!elton.isPlaying()) {
elton.play(ELTON_SAMPLE);
is_playing_compensation = millis() + 100; // See above comment
}

// Need to re-read the LDR values in here when in ELTON mode.
storeLDRRatios();

////////////////////
// ELTON Effect //
////////////////////

// Effects control section. Only active after the GRACE_PERIOD_TIME.
if (millis() > grace_period) {
reverbControl(REVERB_LDR); // Reverb control
filterControl(HP_LDR); // High-pass filter control.
freeze_enabled = granularControl(freeze_enabled, FREEZE_LDR, LENGTH_LDR); // Granular freeze control and
grain length control.
}

///////////////////
// ELTON LEDs //
///////////////////

// The basic "Disco" effect of ELTON mode. Sets the basic light pattern.
eltonDiscoLEDs(wheel_pos);

if (millis() > grace_period) {
freezeLEDsControl(FREEZE_LDR, LENGTH_LDR, blink_state, freeze_enabled);
// Wash color effect for reverb and high-pass.
washColorEffect(Wheel((slow_wheel_pos) & 255), HP_LDR, REVERB_LDR); // Reverb & high-pass effect
}

// Timing control of the basic ELTON disco mode.
if (millis() - last_rainbow_tick >= RAINBOW_INTERVAL){
wheel_pos++;
last_rainbow_tick = millis();
}
// Write all the colors we've been calculating to the pixel but wait to display them.
commitEltonColors();

// Show all the pixels we've been setting.
pixels.show();
////////////////////////////////
// ELTON LED Timing stuff //
///////////////////////////////

// Timing control of the washed out colours.
if (millis() - slow_wheel_tick >= RAINBOW_INTERVAL * 8){
slow_wheel_pos++;
slow_wheel_tick = millis();
if (slow_wheel_pos > 255){
slow_wheel_pos = 0;
}
}

// Time control for the freeze LED blink effect.
if (!freeze_enabled){
blink_interval = getFreezeLength(LENGTH_LDR);
}
if(millis() - blink_time >= blink_interval){
blink_state = !blink_state;
blink_time = millis();
}
}
}
else {
}
// Remember to reset all the effects. Sometimes they linger on startup which is strange during the grace_period.
}

void freezeLEDsControl(int freeze_ldr_n, int length_ldr_n, bool blink_state, bool freeze_enabled){
if (!freeze_enabled){
// Return if freeze is not active. Decided in the audio part.
return ;
}
long blink_color;
if (blink_state){
blink_color = mixColors(pixels.Color(255, 0, 0), pixels.Color(128, 0, 128), ldr_ratios[FREEZE_LDR]);
}
else {
blink_color = mixColors(pixels.Color(0, 0, 255), pixels.Color(128, 0, 128), ldr_ratios[FREEZE_LDR]);
}
for (int k = 0; k < pixels.numPixels(); k++){
setEltonPixelColor(k, blink_color);
}
}

////////////////////////////
// Functions below here //
////////////////////////////

// Reverb Control
void reverbControl(int ldr_n){
verbR.roomsize(ldr_ratios[ldr_n]);
verbL.roomsize(ldr_ratios[ldr_n]);
//verbL.damping(ldr_ratios[ldr_n]);
//verbR.damping(ldr_ratios[ldr_n]);
setVerbDW(volume_scaling(ldr_ratios[ldr_n]));
}

// Reverb DRY/WET balance
void setVerbDW(float dw){
// Maybe response is not always linear!
verbMixR.gain(0, dw); // WET Channel R
verbMixR.gain(3, 1.0 - dw); // DRY Channel R
verbMixL.gain(0, dw); // WET Channel L
verbMixL.gain(3, 1.0 - dw); // DRY Channel L
}

// High-pass filter control
void filterControl(int ldr_n){
filterR.frequency(int(ldr_ratios[ldr_n] * FILTER_MAX));
filterL.frequency(int(ldr_ratios[ldr_n] * FILTER_MAX));
}

// Get the length of the freeze effect.
int getFreezeLength(int length_ldr_n) {
// Needs more testing
return 240 - int(200*ldr_ratios[length_ldr_n]);
}
```

```
bool granularControl(bool freeze_enabled, int freeze_ldr_n, int length_ldr_n){
 bool freeze_event = ldr_ratios[freeze_ldr_n] > GRANULAR_FREEZE_THRESHOLD; // Need to test this value.
 if(freeze_event and !freeze_enabled){
 granR.beginFreeze(getFreezeLength(length_ldr_n));
 granL.beginFreeze(getFreezeLength(length_ldr_n));
 granR.setSpeed(1.0);
 granL.setSpeed(1.0);
 freeze_enabled = true;
 }
 else if (!freeze_event and freeze_enabled){
 granR.setSpeed(1.0);
 granL.setSpeed(1.0);
 granR.stop();
 granL.stop();
 freeze_enabled = false;
 }
 else if(freeze_event and freeze_enabled){
 // Use ldr_ratios to set the playback speed of the grain.
 if (ldr_ratios[freeze_ldr_n] < 0.7) { // Need to test the ratios here further.
 granR.setSpeed(1.0 - (1.0 - ldr_ratios[freeze_ldr_n])*0.3); //  Need to test the ratios here further.
 granL.setSpeed(1.0 - (1.0 - ldr_ratios[freeze_ldr_n])*0.3); // Need to test the ratios here further.
 }
 else {
 granR.setSpeed(1.0);
 granL.setSpeed(1.0);
 }
 }
 return freeze_enabled;
}

// Set the volume of each channel, R/L, of the output mixer. Should only be used in setup.
void setOutMixChannel(int channel, float volume_ratio){
 outMixR.gain(channel, volume_ratio);
 outMixL.gain(channel, volume_ratio);
}
// Automatically set the volume of all the heart beat samples. Reads the ldr_ratios directly as they stand.
void setBeatMix() {
 for ( int k = 0; k < 4; k++){
 beatMix.gain(k, volume_scaling(ldr_ratios[k]));
 }
}
// Mute all the heart beats. Used in ELTON mode.
void muteBeatMix() {
 for ( int k = 0; k < 4; k++){
 beatMix.gain(k, 0.0);
 }

}

// Read the LDR pins and return them, called from storeLDRRatios(). They are then converted to ratios.
void rawLDRValues(int *values){
 for(int k = 0; k < 4; k++){
 values[k] = analogRead(LDR_PINS[k]);
 }
 if (DEBUG){
 for(int k = 0; k < 4; k++){
 Serial.print(values[k]);
 Serial.print("\t(");
 Serial.print(ldr_ratios[k]);
 Serial.print(")\t");
 }
 Serial.println();
 }
}
// Convert the LDR values (read using the rawLDRValues() function) to ratios and store them in memory.
void storeLDRRatios(){
 int raw_values[] = {0, 0, 0, 0};
 rawLDRValues(raw_values);
 for(int k = 0; k < 4; k++){
 // min(max()); in order to constrain values in case of failed ambient readings.
 //ldr_ratios[k] = min(max(float(raw_values[k] - LDR_0[k]) / float(LDR_M[k]), 0.0), 1.0);
 float new_value = min(max(float(raw_values[k] - LDR_0[k]) / float(LDR_M[k]), 0.0), 1.0);
 // Some sort of "glide" implementation. Did not work well.
// if (ldr_ratios[k] - new_value > 0.02){
// new_value = ldr_ratios[k] - 0.01;
// }
// else if (ldr_ratios[k] - new_value < 0.02){
// new_value = ldr_ratios[k] + 0.01;
// }
 ldr_ratios[k] = min(max(new_value, 0.0), 1.0);
 }
}
```

```
// Checks if ELTON mode should be triggered. The for loop uses fancy programming math to check if all the ldr_ratios
are above the threshold.
bool triggerElton(){
 // Returns true if all LDR values are above the TRIGGER_ELTON_THRESHOLD.
 bool condition = true;
 for (auto element : ldr_ratios){
 condition &= element > TRIGGER_ELTON_THRESHOLD;
 }
 return condition or DEBUG_ELTON;
}

// Checks if we should remain in ELTON mode.. The for loop uses fancy programming math to check if any the
ldr_ratios are above the threshold.
bool remainElton(){
 // Returns true if any LDR value is above the REMAIN_ELTON_THRESHOLD.
 bool condition = false;
 for (auto element : ldr_ratios){
 condition |= element > REMAIN_ELTON_THRESHOLD;
 }
 return condition or DEBUG_ELTON;
}

// Scale the ldr_ratio values for controling volume. This is done to quickly (but smoothly) bring the volume to max, since
subtle mixing isn't interesting here.
float volume_scaling(float value_in) {
 if (value_in < BEAT_VOL_MIN_THRESHOLD) {
 return 0.0;
 }
 else if (value_in > BEAT_VOL_MAX_THRESHOLD) {
 return 1.0;
 }
 else {
 float vol_val = (value_in - BEAT_VOL_MIN_THRESHOLD) / (BEAT_VOL_MAX_THRESHOLD -
BEAT_VOL_MIN_THRESHOLD);
 return vol_val;
 }
}

////////////////////
// LED functions //
////////////////////

// The basic LED pattern of the unaffected ELTON mode. LEDs cycling through nice colours at a nice speed.
void eltonDiscoLEDs(int wheel_pos){
 for (int k=0; k<pixels.numPixels(); k++){
 long colour = Wheel(((k * 256 /pixels.numPixels()) + wheel_pos) & 255);
 setEltonPixelColor(k, color);
 }
}

// The reverb and high-pass filter LED effects combo.
void washColourEffect(long slow_colour, int hp_ldr_n, int rev_ldr_n){
 float mix = max(ldr_ratios[hp_ldr_n], ldr_ratios[rev_ldr_n]); // Use the higher of the LDR values to control the
magnitude of the effect.
 float white_mix = ldr_ratios[hp_ldr_n]; // How desaturated is the final colour?
 uint8_t wash_white = (1.0 - white_mix * WHITE_WASH_RATIO) * 255; // Maintain even brightness since white is just
turning all the colours on.
 long wash_colour = mixColours(pixels.Colour(wash_white, wash_white, wash_white), slow_colour, white_mix); //
create the washed base colour
 for (uint8_t k = 0; k < pixels.numPixels(); k++){
 long colour = mixColours(wash_colour, pixels.Colour(COLOURS[k][0], COLOURS[k][1], COLOURS[k][2]), mix); // Mix
the washed colour with the basic "disco" mode colour.
 setEltonPixelColor(k, color);
 }
}

// Set pixel colours in an array, COLOURS, before committing them to the pixels themselves. This is for having a
dynamic colour effect that can be affected.
void setEltonPixelColour(int n, long colour){
 COLORS[n][0] = getRed(color);
 COLORS[n][1] = getGreen(color);
 COLORS[n][2] = getBlue(color);
}

// Commit the ELTON mode colours to the pixels but don't display them. Use pixels.show() for that.
void commitEltonColours(){
 for (int k = 0; k < pixels.numPixels(); k++){
 pixels.setPixelColor(k, pixels.Color(COLORS[k][0], COLORS[k][1], COLORS[k][2]));
 }
}
```

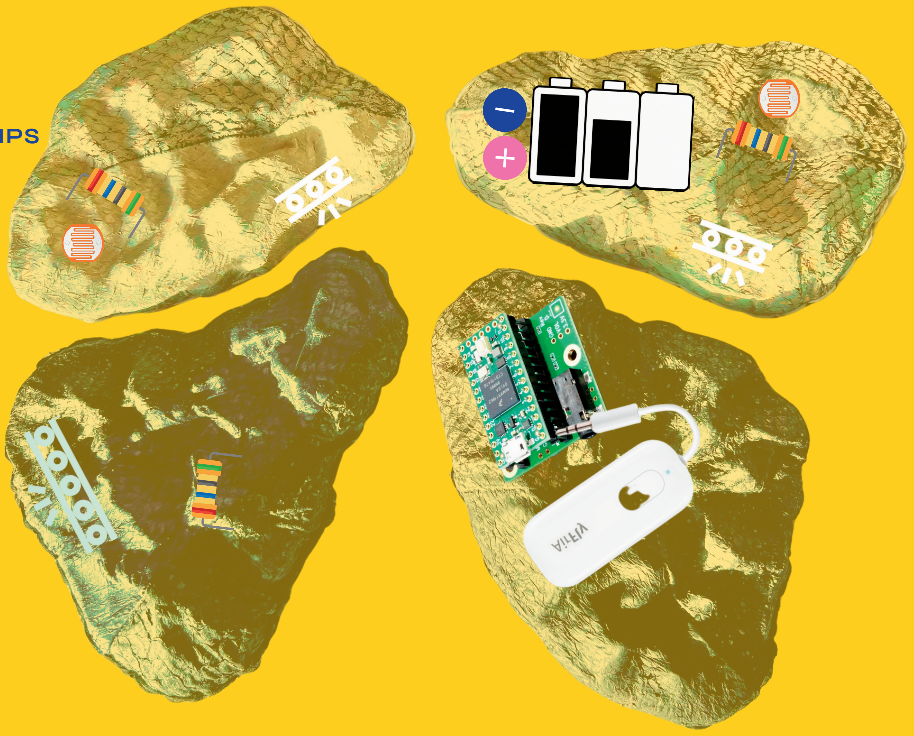
```
// Blink all the LEDs as needed in time with the heart beats. This is run from the main program.
void blinkHeartBeatLEDs(){
 for (int k = 0; k < 4; k++){
 heartBeatLED(k);
 }
}

// Control each LED group by detecting the heart beats.
void heartBeatLED(int n){
 if (peaks[n].available()){
 float p = peaks[n].read();
 uint32_t c = pixels.Color(255 * p * ldr_ratios[n], 0, 0, 0);
 setPixelGroupColour(n, c);
 pixels.show();
 }
}

// Set the colour of each pixel in a group at the same time. The groups are defined in the global pixelGroups variable.
void setPixelGroupColour(int n, int c){
 for (int k = 0; k < pixels.numPixels(); k++)
 {
 if(pixelGroups[k] == n){
 pixels.setPixelColour(k, c);
 }
 }
}

// Colour Wipe. Only used at startup. Used to indicate the calibration status.
void colourWipe(uint32_t c, uint16_t wait)
{
 for(uint16_t i=0; i<pixels.numPixels(); i++)
 {
 pixels.setPixelColour(i, c);
 pixels.show();
 delay(wait);
 }
}

// Input a value 0 to 255 to get a colour value.
// The colours are a transition r - g - b - back to r.
// Borrowed directly from the BrightDot example program.
uint32_t Wheel(byte WheelPos)
{
 if(WheelPos < 85)
 {
 return pixels.Colour(WheelPos * 3, 255 - WheelPos * 3, 0);
 } else if(WheelPos < 170)
 {
 WheelPos -= 85;
 return pixels.Colour(255 - WheelPos * 3, 0, WheelPos * 3);
 }
 else
 {
 WheelPos -= 170;
 return pixels.Colour(0, WheelPos * 3, 255 - WheelPos * 3);
 }
}

// Mix 2 colours to achieve a linear combination.
long mixColours(long c1, long c2, float mix){
 uint8_t r = getRed(c1) * mix + getRed(c2) * (1.0 - mix);
 uint8_t g = getGreen(c1) * mix + getGreen(c2) * (1.0 - mix);
 uint8_t b = getBlue(c1) * mix + getBlue(c2) * (1.0 - mix);
 return pixels.Colour(r, g, b);
}

// Extract individual colours from a full colour. This is used to combine multiple LED effects in ELTON mode
uint8_t getRed(long colour){
 return (uint8_t)((colour >> 16) & 0xff);
}

// Extract individual colours from a full colour. This is used to combine multiple LED effects in ELTON mode
uint8_t getGreen(long colour){
 return (uint8_t)((colour >> 8) & 0xff);
}

// Extract individual colours from a full colour. This is used to combine multiple LED effects in ELTON mode
uint8_t getBlue(long colour){
 return (uint8_t)(colour & 0xff);
}
```

AirFly

THE CIRCUIT

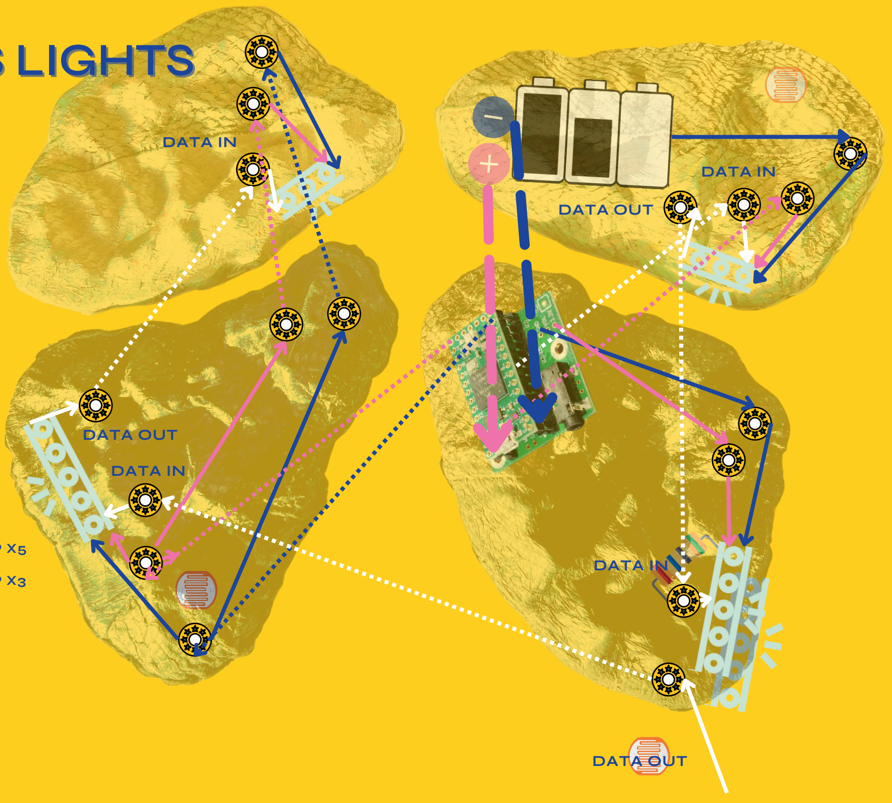
HARDWARE

- 4 x BRIGHTDOT STRIPS
- TEENSY 4.0 + AUDIO SHIELD
- 3 x AA BATTERY
- 4 x LDRs
- AIRFLY PRO BLUETOOTH TRANSMITTOR



CONNECTIONS LIGHTS

- POPPERS
- INTERNAL BRIGHTDOT DATA
- EXTERNAL KUMIHIMO BRIGHTDOT DATA
- INTERNAL GROUND
- EXTERNAL KUMIHIMO WIRE GROUND
- INTERNAL 3.3V POWER
- EXTERNAL KUMIHIMO WIRE 3.3V POWER
- MAIN 5V POWER IN
- MAIN GROUND
- 2 x VELLEMAN BRIGHTDOT STRIP X5
- 2 x VELLEMAN BRIGHTDOT STRIP X3

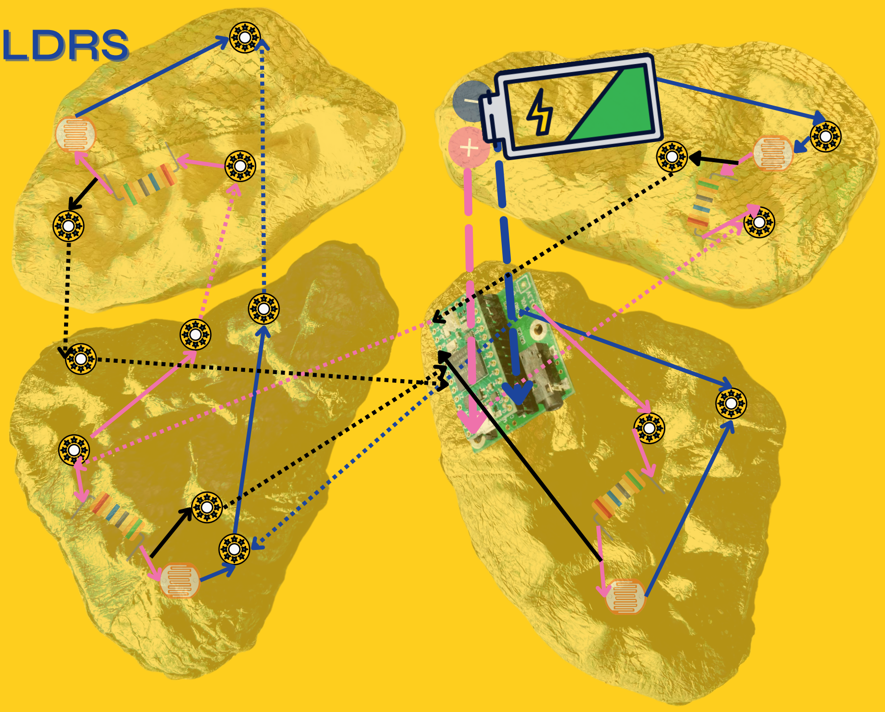


ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

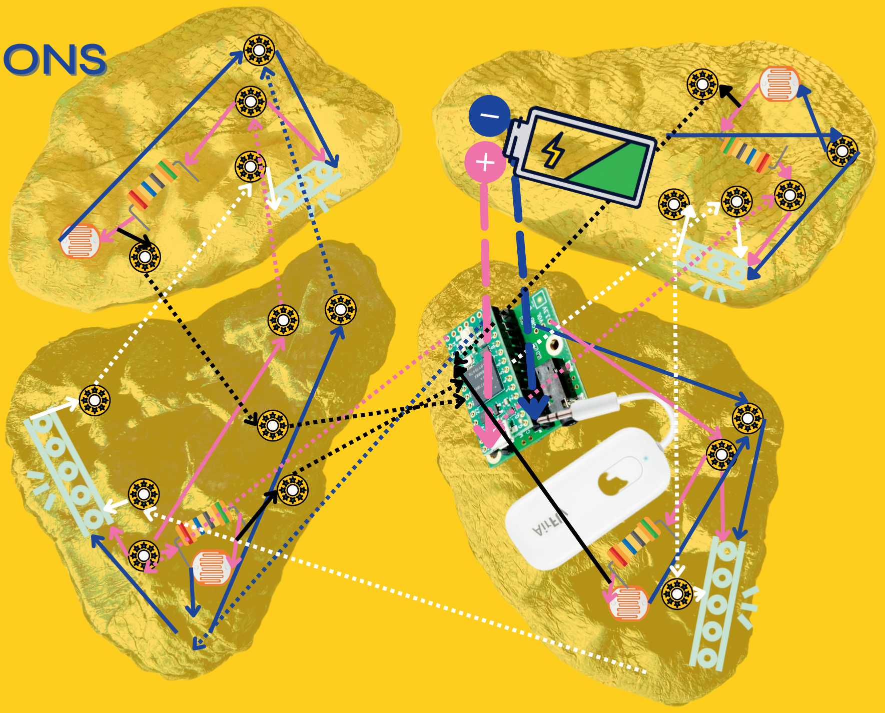
THE CIRCUIT

CONNECTIONS LDRS

●	POPPERS
→	INTERNAL LDR DATA
⋯→	EXTERNAL KUMIHIMO LDR DATA
→	INTERNAL GROUND
⋯→	EXTERNAL KUMIHIMO WIRE GROUND
→	INTERNAL 3.3V POWER
⋯→	EXTERNAL KUMIHIMO WIRE 3.3V POWER
➔	MAIN 5V POWER IN
➔	MAIN GROUND
▤	4 x LDRs

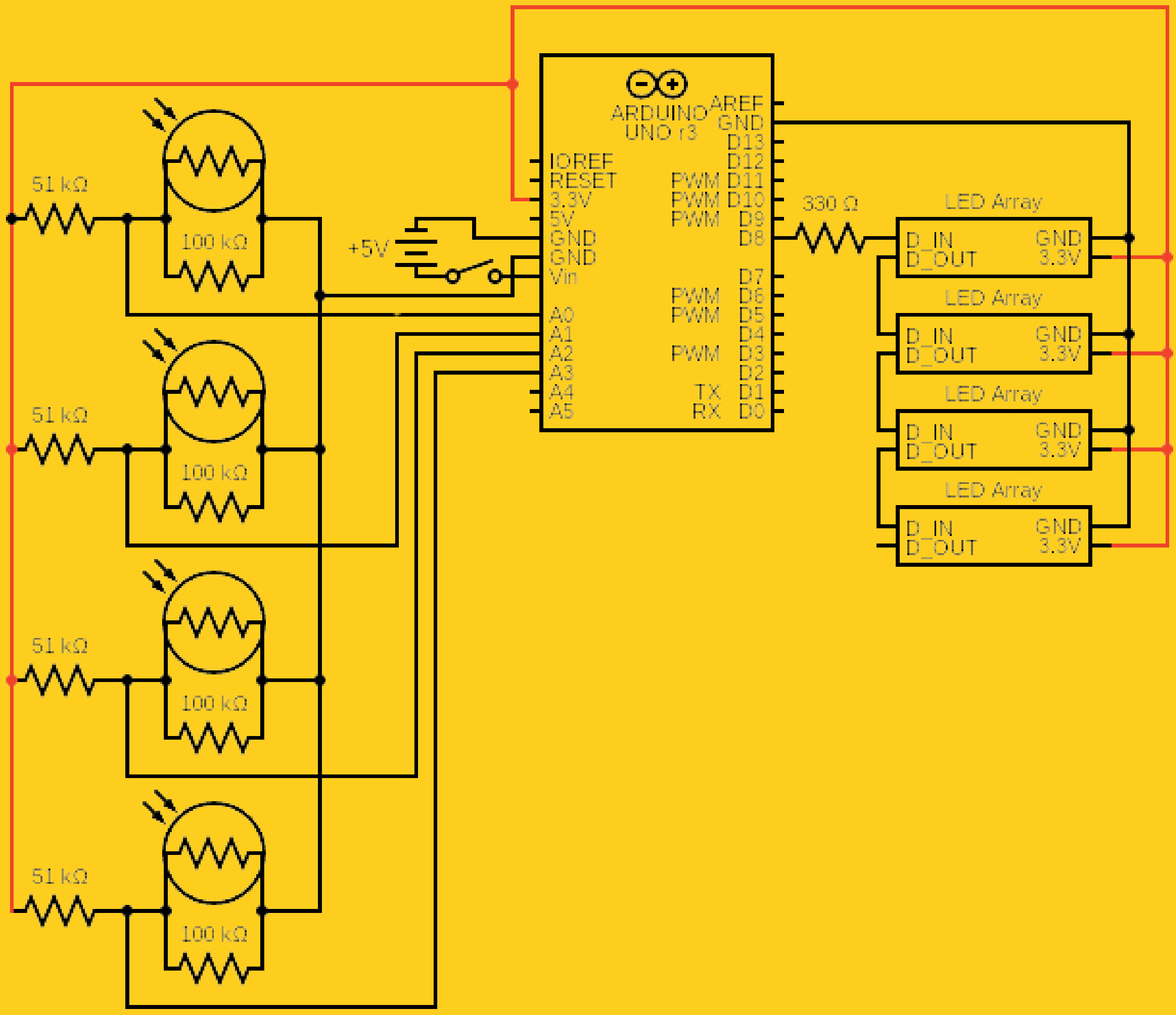


ALL CONNECTIONS



THE CIRCUIT

WIRING SCHEMATIC



CONNECTIONS

As I did not want the hearty party to look electronic I started experimenting with ways of creating the connections or ateries between the chambers of the heart but soft.

I heard about the technique of Kumihimo braiding using a disc to braid cords. I 3D printed a kumihimo disc file I found online. I then made various prototypes trapping conductive thread inside different braid patterns up to 32 threads.

I eventually decided to use 0.65mm 22 American wire trapped inside a 4 thread braid made in the most simple cross configuration. Using diferent colours of blue black and silver in leftover embroidery threads and Icelandic eimband yarn.

After the braids were made they were either soldered directly onto the Teensy or onto a popper that would coresspond to a connecting popper connected inside the soft circuit itself. Therefore these are easily connected and disconnected. One of theres wires has been braided around an on and off switch which is visable on the front on purpose. This physical interation allows the user to be in control of the callibration mode that is essential for use in different ambient light conditions.

.

ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

CONCLUSION

Through trial and many errors I feel I have accomplished a very nice project. For me the greatest achievement was giving the piece to my fellow heart attack survivor women from Reykjalundur to test in the project video. They loved it and got the point of it being so much fun to play. This was the best part of the completion for me. I hope that this video and my research will at least make one woman aware of the issues surrounding heart disease and my work will all be worthwhile.

There are various things that I feel can be improved on now I have more knowledge and experience working in these fields. In particular the circuit design could of been a bit more thought through in terms of embedding. Its the kind of thing that you don´t really know until you do it more frequently.

I have also learned lots about music and music theory through this project that I feel will enhance my passion for sound.

This whole experience has been a very therapeutic way of recovering from my health issues both mentally and physically. I now have gone from a pretty computer illiterate person to programming and designing 3D models in 6 months.

I feel that the Fabricademy experience has given me the ability to achieve anything I can put my mind to. If I have a creative problem I know through research that I can find how to fix that problem and probably make the tool I need to fix it myself.

I hope that you have enjoyed reading about the hearty party as much as I enjoyed making it!

ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy

THE FUTURE

I am really looking forward to working on some new work with all of my new skills. I hope to further explore fish leather and moulding. I also have a new love for e-textiles that I hope to continue and expand on. I also feel there is a lot of interesting things that can be made with 3D printed jewellery and fabrics. These areas are not really being looked at within Iceland so there id a market for new designs here.
I might have a job in the circular economy here where I can keep improving my skills and knowledge base. I would love to pass on all that I´ve learned to others here. I fully believe in open source and skill sharing and changing the world.

Heres to the future!

ACKNOWLEGEMENTS

THE HEARTY PARTY WOULD NOT OF BEEN POSSIBLE WITHOUT THE RECEIVING A SCHOLARSHIP FROM CENTRINNO EU.

THANKS TO THE AMAZING LOUISE MASSACRIER WHO HAS TAUGHT ME SO MUCH AND TO MY FABRICADEMY FAMILY AT THE ICELANDIC TEXTILE CENTER. ALBERTE, ALICE, MARGRÉT AND ARIANE.

ALSO THANKS TO MY PARTNER DETTIFOSS BERGMANN FOR ALL THE SUPPORT AND TALENT. THANKS FOR TEACHING ME MUSIC THEORY AND PROGRAMMING.

BIG SHOUT OUT TO...

LINDA WALKER BOLLOM + ANDRÉA GUÐMUNDSDÓTTIR THE REYJALUNDUR MASSIVE AND BEST MODELS EVER.

REYKJALUNDUR FOR GETTING ME TO A PLACE WHERE I COULD DO THIS.

THE INTELLIGENT INSTRUMENTS LAB.

DIANE WAKIM FOR THE SUPPORT

ÞORBJÖRN GUÐJÓNSSON MY BRILLIANT CARDIOLOGIST

CAROLINA SALAS FOR FILM ADVICE AND LENDING ME THE LIGHTS

HARPA. IN REYKJAVIK FOR ALLOWING THE FILMING OF THE PROJECT.

TOO ALL MY SUPPORTIVE FRIENDS THAT HAVE KEPT ME SANE.

RESOURCES

All of my documentation and process can be found online by scanning this QR Code.

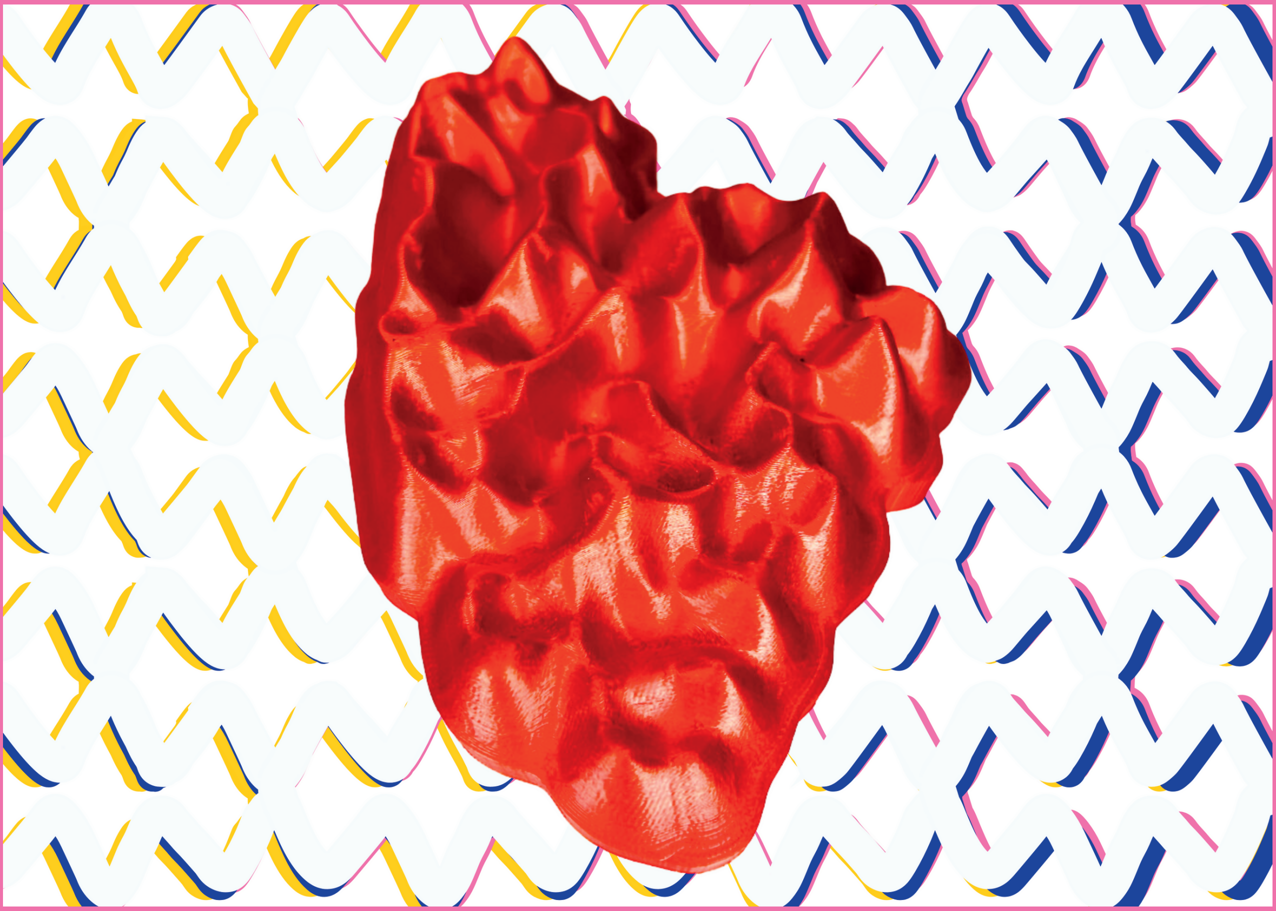
There you will also find all the links to all the research contained within this booklet. I am happy for anyone to contact me regarding this project or any of the work I have made throughout my Fabricademy journey.

I hope that anyone would be able to follow the process and be inspired by it themselves.

emmashannon1@gmail.com

winkstex

00354 7810232



THE HEARTY PARTY by Emma Shannon

THE HEARTY PARTY IS A WEARABLE SYNTHESIZER
INSPIRED BY THE HEART

THIS PROJECT CREATES A DIALOGUE BETWEEN
WEARABLE TECH, PARAMETRIC DESIGN AND
TRADITIONAL CRAFT TO RAISE AWARENESS OF HEART
DISEASE IN A FUN AND PLAYFUL WAY!

ICELANDIC TEXTILE CENTER
CENTRINNO
Co-funded by the Creative Europe Programme of the European Union
FABRICADEMY textile and technology academy