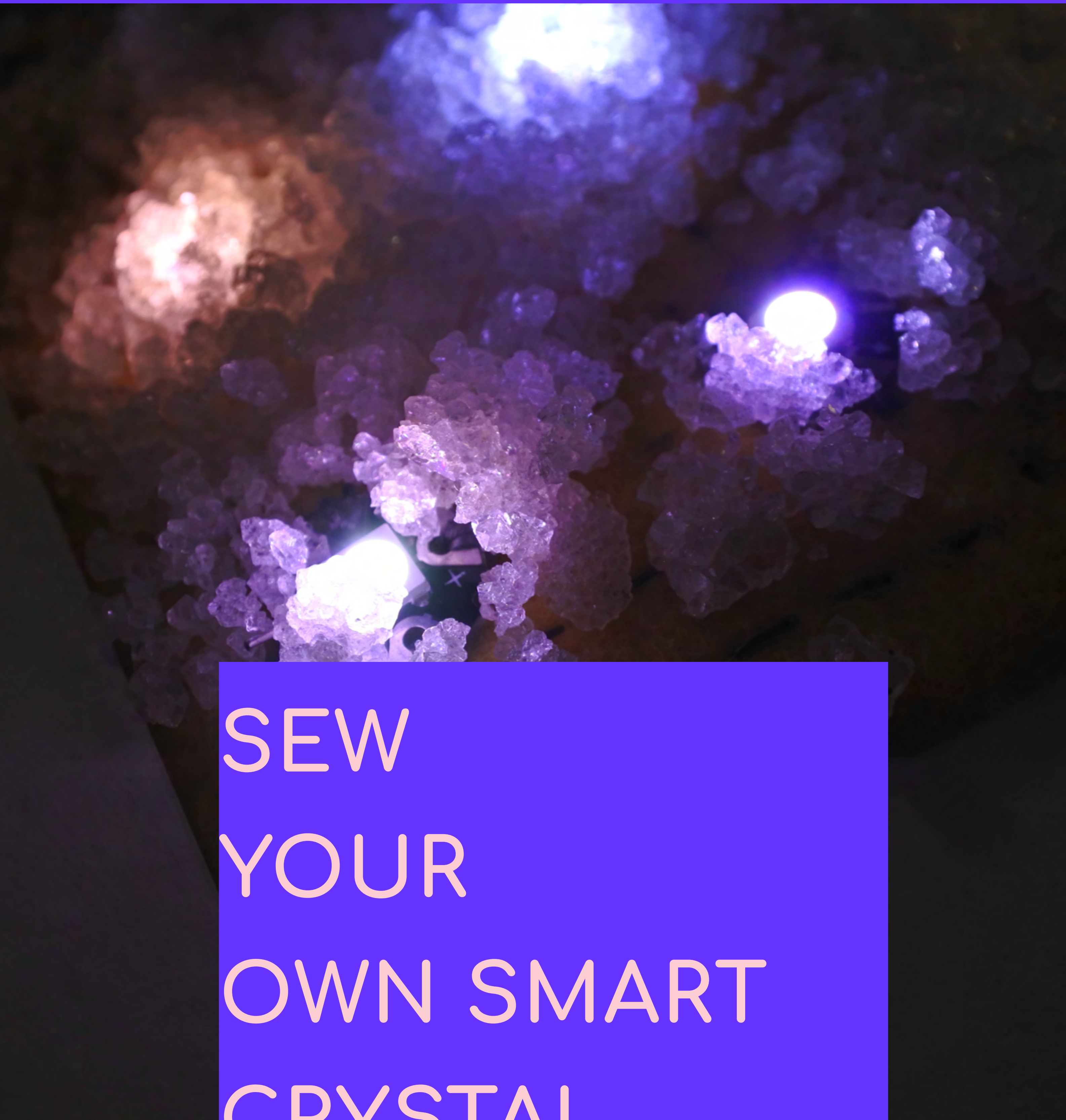
A MAKER PEDAGOGICAL RESOURCE



SEW YOUR OWN SMART CRYSTAL LAMP

A PUBLICATION BY ONL'FAIT

SEW YOUR OWN SMART LAMP!
A maker pedagogical resource
Author : Cristina Olivotto
Editor : Onl'Fait
Drawings and images : Cristina Olivotto
Editing: Sirius Interactive Ltd

Fabricademy

This activity has been developed within the framework of the first edition of Fabricademy – a 6-month transdisciplinary course that focuses on the development of new technologies in the textile industry. The course covers a broad range of textile applications – from fashion to the upcoming trend of wearable technology. Currently, the global textile and clothing industry is exploring and experimenting with more viable, sustainable and fair production systems. Fabricademy's programme highlights the unethical and environmentally unfriendly practices of the current industry and strives to combine traditional knowledge and future craftsmanship to work towards new ways of designing, prototyping and producing for the newly emerging fashion and textile industry.

textile-academy.org

Onl'Fait

The current final project has been designed and developed at Onl'Fait, the first pedagogical Fab Lab in Geneva, Switzerland. Onl'Fait is an open educational-technological maker space, where everyone – artisans, children, artists, scientists – is welcome to learn, make, share, fix, create, discuss, teach, fail, tinker and reflect on a sustainable approach to science & technology. The objectives of Onl'Fait are to:

- promote the spirit and mindset shared by scientists and makers: analyse, research, create, modify, solve
- encourage inclusion in our community: from children to scientists. Onl'Fait contributes to sharing scientific and technological knowledge in innovative ways
- facilitate science learning by offering hands & minds-on programmes, especially for children
- support the open hardware and software movement to share, own and contribute to science & technology knowledge and culture.

onlfait.ch

How can you combine electronics, chemistry and sewing in a single activity?
By sewing a smart crystal lamp! You need to use a needle and thread to stitch the electronic components, code an Arduino microcontroller to make your lamp responsive to temperature and finally decorate it with home-grown aluminium crystals.
Ready?

Outline

In this interdisciplinary activity, students will build a smart crystal lamp by sewing the components together. Students will work in pairs or groups of 3 to build the lamp in four steps. (1) They will sew an electronic circuit; (2) they will code an Arduino microcontroller and connect it to the sewn circuit and to a temperature sensor; (3) they will assemble the lamp and test it; and (4) they will grow crystals on the lamp to enhance its aesthetic appeal.

Curriculum

- Technology
  - Electronics
    - Microcontrollers
    - Power supplies
    - LEDs
    - Sensors
    - Resistances
  - Coding
    - Arduino language (C/C++)
- Science
  - Chemistry
    - Crystal growth
  - Physics
    - Electric circuits
- Art & design
  - Sewing
  - Prototyping
  - Making

Students will learn

- The basics of electronics
- The basics of coding
- The process of crystal growth
- The process of prototyping
- Team work

A.

LEARNING BY MAKING

The concept of 'learning by making' is rooted in constructivism – a theory of learning based on experience, observation and reflection, developed by Swiss psychologist Jean Piaget. In constructivist education settings, students are offered authentic, real-life learning opportunities, and they learn by making tangible objects. These opportunities allow for a guided, collaborative process that incorporates peer feedback. The 'learning by making' approach places a strong emphasis on the development of positive technological fluency in students and the promotion of learning through design and sharing in collaborative environments (Papert, 1996). The efficacy of 'learning by making' has been confirmed among diverse student populations, with a range of abilities. It is a gender-inclusive method of science education that enhances participation and interest in science activities (Rocard et al, 2007).

The maker movement is a growing movement of 'hobbyists, tinkerers, engineers, hackers, and artists committed to creatively designing and building material objects for both playful and useful ends' (Martin, 2015). It has spread rapidly since the founding of Make magazine in 2005 – an initiative of Dale Dougherty, who describes the maker movement as 'people's need to engage passionately with objects in ways that make them more than just consumers' (Dougherty, 2012). Today, the maker movement has been globally recognised as being capable of transforming the fields of science, technology, engineering, mathematics (STEM). It can be viewed as a response to a call for better student engagement in education. For instance, Obama (2009) said, "I want us all to think about new and creative ways to engage young people to create and build and invent to be makers of things, not just consumers of things'. Constructivist theories are sometimes considered to underpin the learning that occurs by makers' focus on production and problem solving (Harel & Papert, 1991). According to Stager (2014), 'The maker movement is a vehicle that will allow schools to be part of the necessary return to constructivist education. A movement that will allow innovative, independent, and technologically literate; not an "alternative" way to learn, but what modern learning should really look like'.

In many schools, teachers do not always have the resources to create a makerspace or they do not have the time to help students investigate problems of their choice or support the development of certain skills. Onl'Fait attempts to address these issues by providing the space, resources, expertise needed for maker-based STEM education as well as the support and training for educators.

Martin (2015) outlines three elements that are essential to help derive the most out of the maker movement in educational settings: (1) digital tools, including rapid prototyping tools and low-cost microcontroller platforms that are needed in many making projects; (2) community infrastructure, including online resources and in-person spaces and events; and (3) a maker mindset, aesthetic principles and mental habits that are commonplace within the community. Onl'Fait can provide all the three elements to schools.

Essential reading

Dougherty D., Conrad A. "Free to Make" (Berkley, North Atlantic Books, 2016).

Harris E., Winterbottom M., Xanthoudaki M., Calcagnini S., De Puer I. "Tinkering: A Practitioner Guide for Developing and Implementing Tinkering Activities" (Open Education Resources, 2017).

Honey M, Kanter D.E. "Growing the Next Generation of STEM Innovators" (New York, Routledge, 2013).

Peppler K., Rosenfeld Halverson E., Kafai Y.B. (eds) "Makeology. Makers as learners" (New York, Routledge, 2016).

Tinkering Studio's MOOC "Tinkering Fundamentals: A Constructionist Approach to STEM Learning"

Fast Facts

Age range: 14–18 years old
Type: Classroom activity
Complexity: High
Teacher preparation time: 3 weeks to order and buy components
Time required: About 16 hours + 1 week waiting time

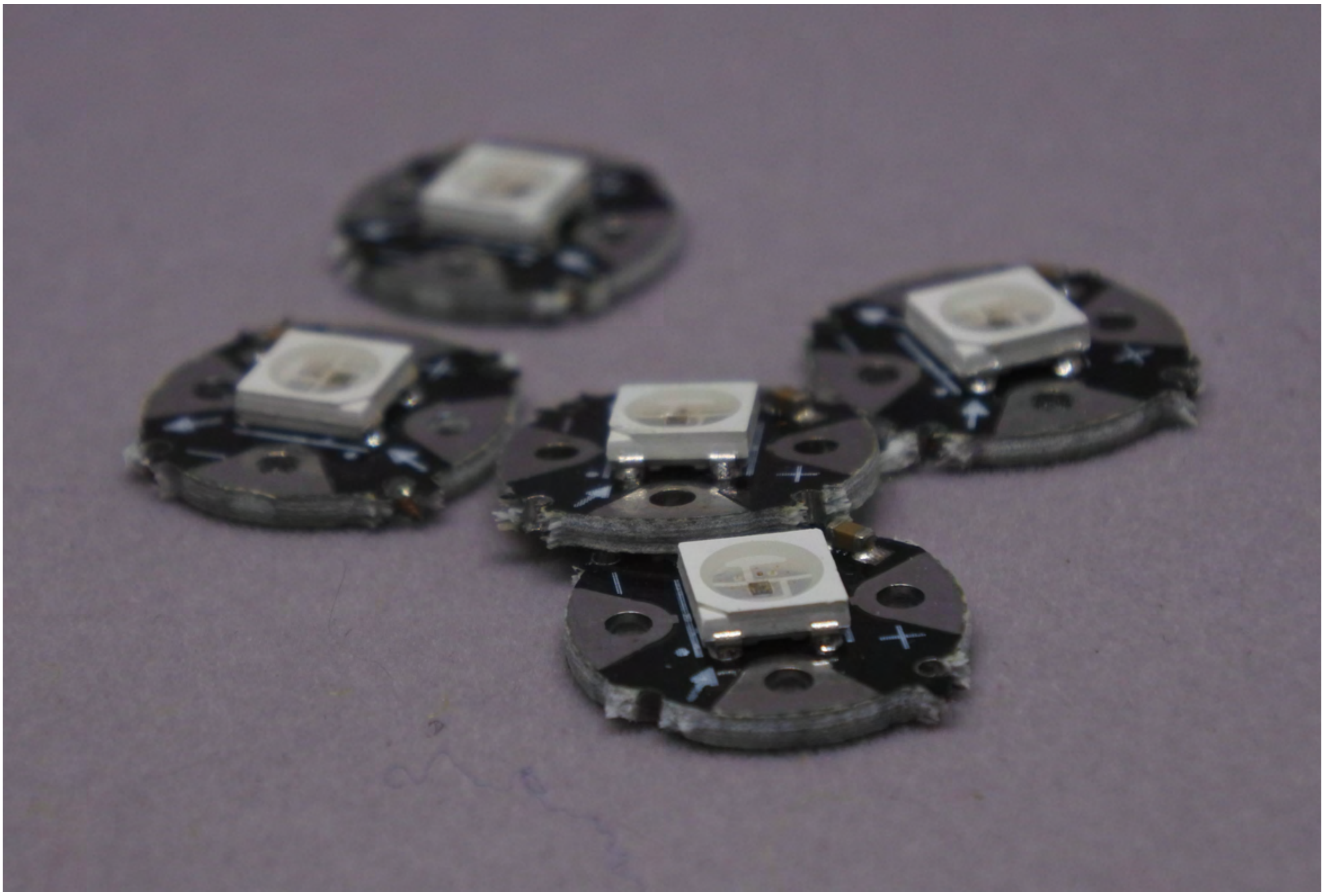
- Sewing: 4 hours
- Hardware: 4 hours
- Coding: 4 hours
- Crystals: 2 hours + 1 week + 2 hours

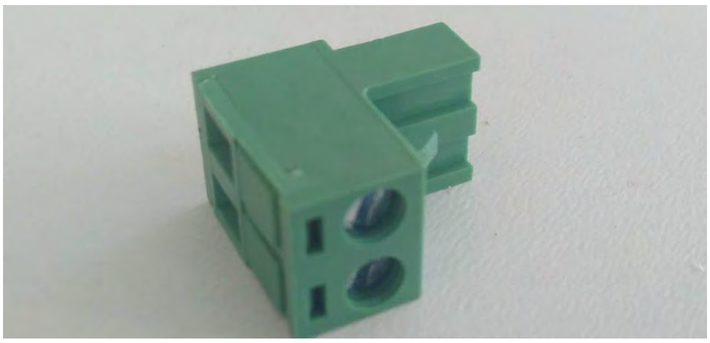
Cost: High
Location: Indoor (any classroom)

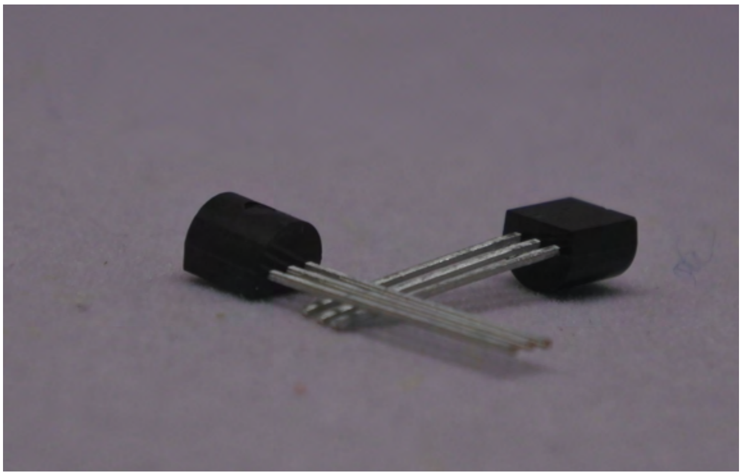
Material

- Thin light-coloured felt (22 cm x 10 cm, 1 mm thick)
- Vinyl glue
- Conductive thread
- Needles
- 12 Neopixel LEDs (2 spares)
- Arduino UNO + USB cable
- Power supply cable for Arduino 6-10V
- Power supply cable for the LEDs 5V 2A
- Wire stripper
- 2 Mammut clamps
- Temperature sensor TMP36
- 10 male – male jumper wires (3 spares)
- 1 small breadboard
- Soldering irons + solder
- Transparent nail polish
- Paper frame Openframe
- Computer
- Potassium alum (700 g)
- Pastic basin 25 cm x 15 cm x 5 cm (about)
- Double-sided tape
- Superglue


UNO
ARDUINO



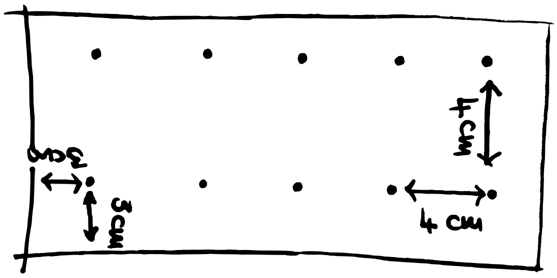




B.

THE SEWN CIRCUIT

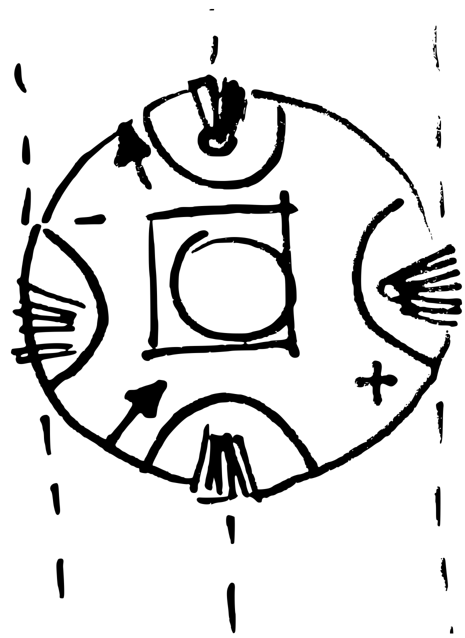
1. Take a 22 x 10 cm piece of felt. Leave 3 cm margin on each side, and with a marker, make a grid of 10 dots, with each dot at a distance of 4 cm from the previous one.



2. Use the vinyl glue to stick 10 Neopixel LEDs on the reference grid. Refer to the drawing below for the orientation of the LEDs so that they are positioned in parallel. In each row, the plus signs should be on the same side and the minus on the opposite side. This way, all the little arrows will point in the same direction. In the second row, this pattern will be reversed.

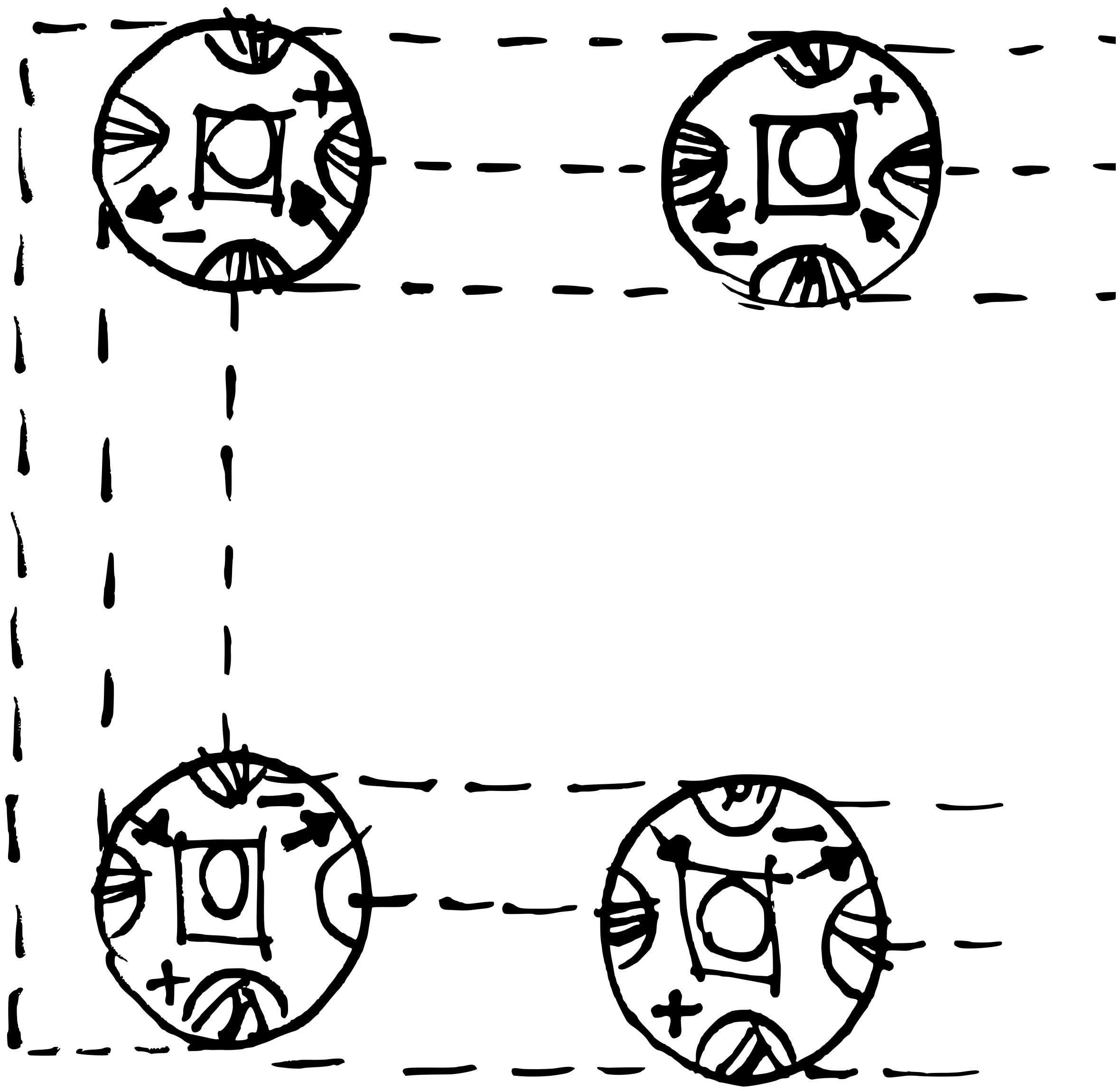
3. Connect all the + terminals. Cut about 1.5 m of conductive thread, and thread a needle. Make a knot to fix the thread to the felt piece and pass the thread through the + terminal 3-5 times. Once the connection is strong, continue with the next LED and connect all the + ends. If the thread is not long enough, pass the thread through the + terminal of the last LED, make a knot in the felt and cut the thread. Start the procedure again from the same LED to make sure that the two threads are tightly bound.

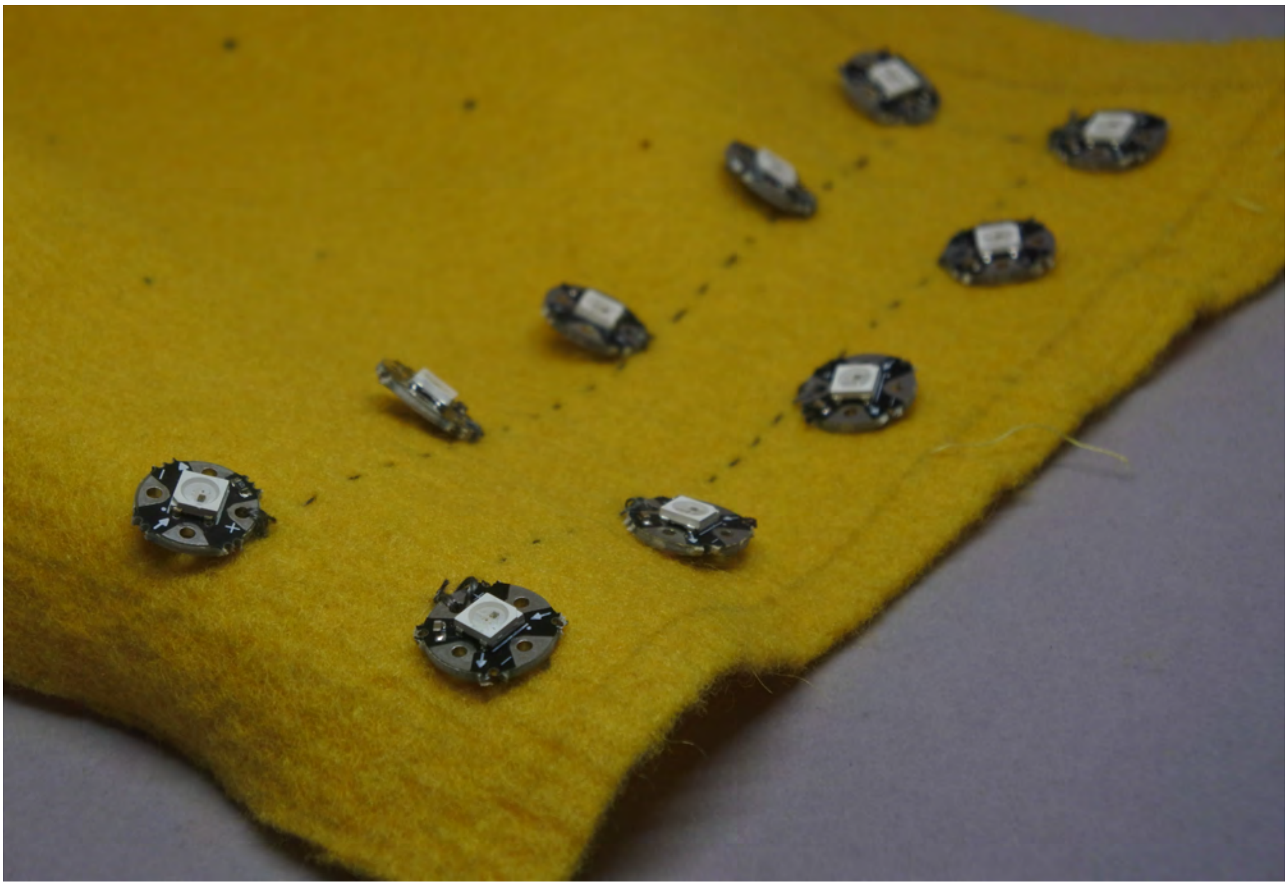
4. Connect all the minus terminals. Follow the same steps

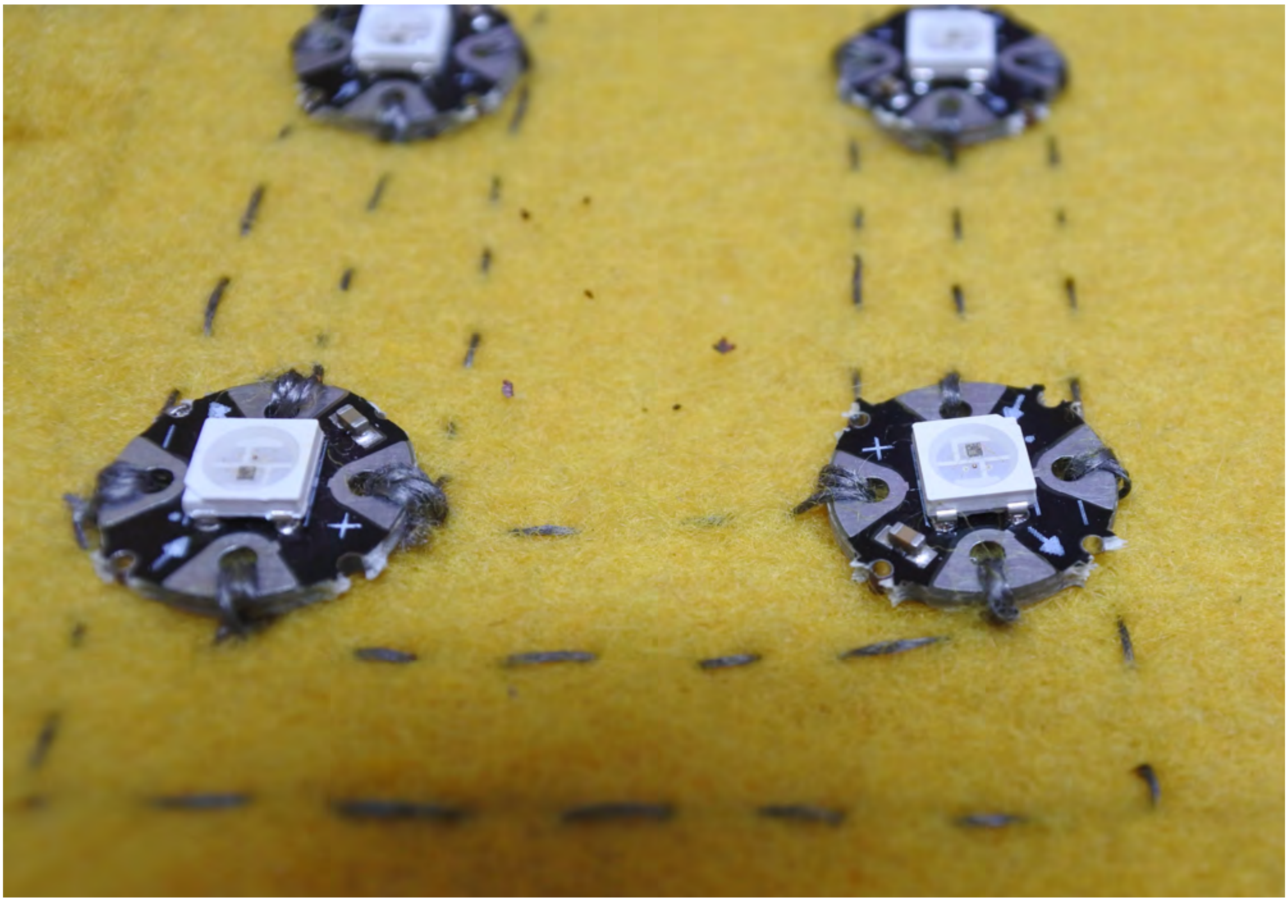


as in point 3. Be careful never to cross or touch the thread connected to the plus terminals.

5. After all the + and - terminals are connected in parallel, you have to connect adjacent LEDs to each other in a series. Cut a small piece of thread. Tie one end of the thread to the data terminal of one LED and the second end to the data terminal of the adjacent LED. Pair all the LEDs in this way (see diagram below).







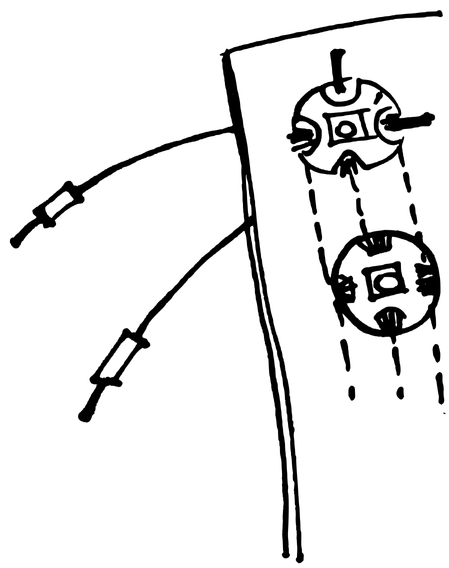
E-TEXTILE

Electronic textiles, also known as smart garments, are fabrics with embedded electronic components such as small computers (microcontrollers), batteries, lights (LEDs), buzzers, etc. Smart fabrics have enormous potential because they have several advantages over traditional fabrics, including the ability to communicate, collect data and transform. Technology can be used to enhance not only the aesthetic appearance of some e-textiles but also their performance. According to the Fabricademy manifesto, e-textiles ‘experiments with the human body, culture and mindset by recycling, hacking and sensing it, creating feedback loops with project development, where materials, aesthetics, sustainability and customization play equal and important roles’.

C.

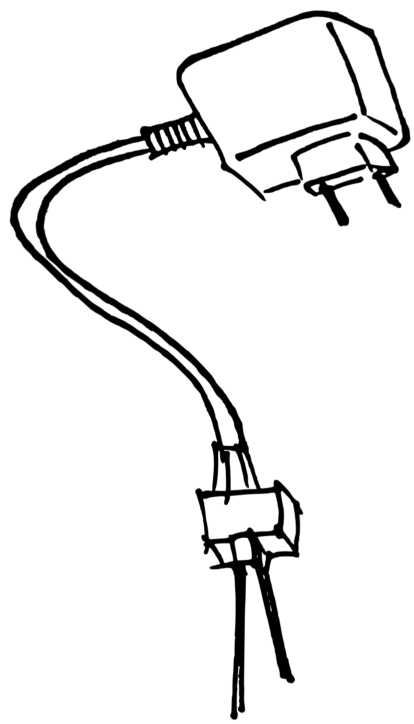
THE CIRCUIT

1. Begin by connecting to the ground. Connect the - pin of the LED series to the Arduino GND with a jumper wire. To make a soft-hard connection, insert the jumper wire into the minus pin of the LED first and solder. Now, use another jumper wire and a mammut clamp to connect to the - pin of the LED series to the - wire of the power supply. Preferably choose an LED at the centre of the grid (e.g. the 5th LED) to make sure that the electricity is evenly distributed



over the circuit. To make a soft-hard connection, insert the jumper wire into the - pin of the LED first and solder it.
Note that the two jumper wires should be connected from the back of the felt piece. The ground of the temperature sensor should be connected to the Arduino GND through a breadboard and a jumper wire.

2. Both the temperature sensor and the pixel work well with 5V power; however, the LEDs series needs about 2 A (see box below), which the Arduino cannot provide. An external power supply is therefore needed. Connect the + pin of the sensor to the Arduino board as shown in



the scheme below (5V pin on Arduino). Connect the + pin of the LEDs series to the + wire of the power supply with a mammut clamp. Again, choose an LED in the centre (the 5th) to make sure the electricity is evenly distributed over the circuit. For a soft-hard connection, insert the jumper wire into the - pin of the LED first and solder it.

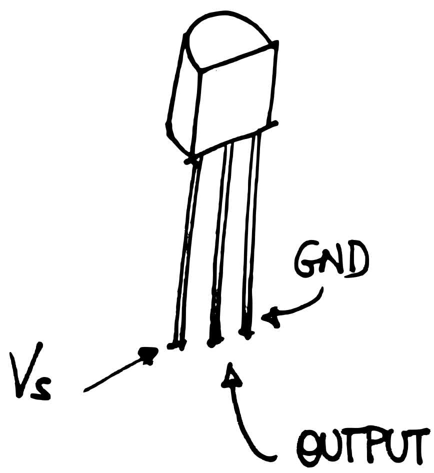
The jumper wires should be connected from the back of the felt piece.

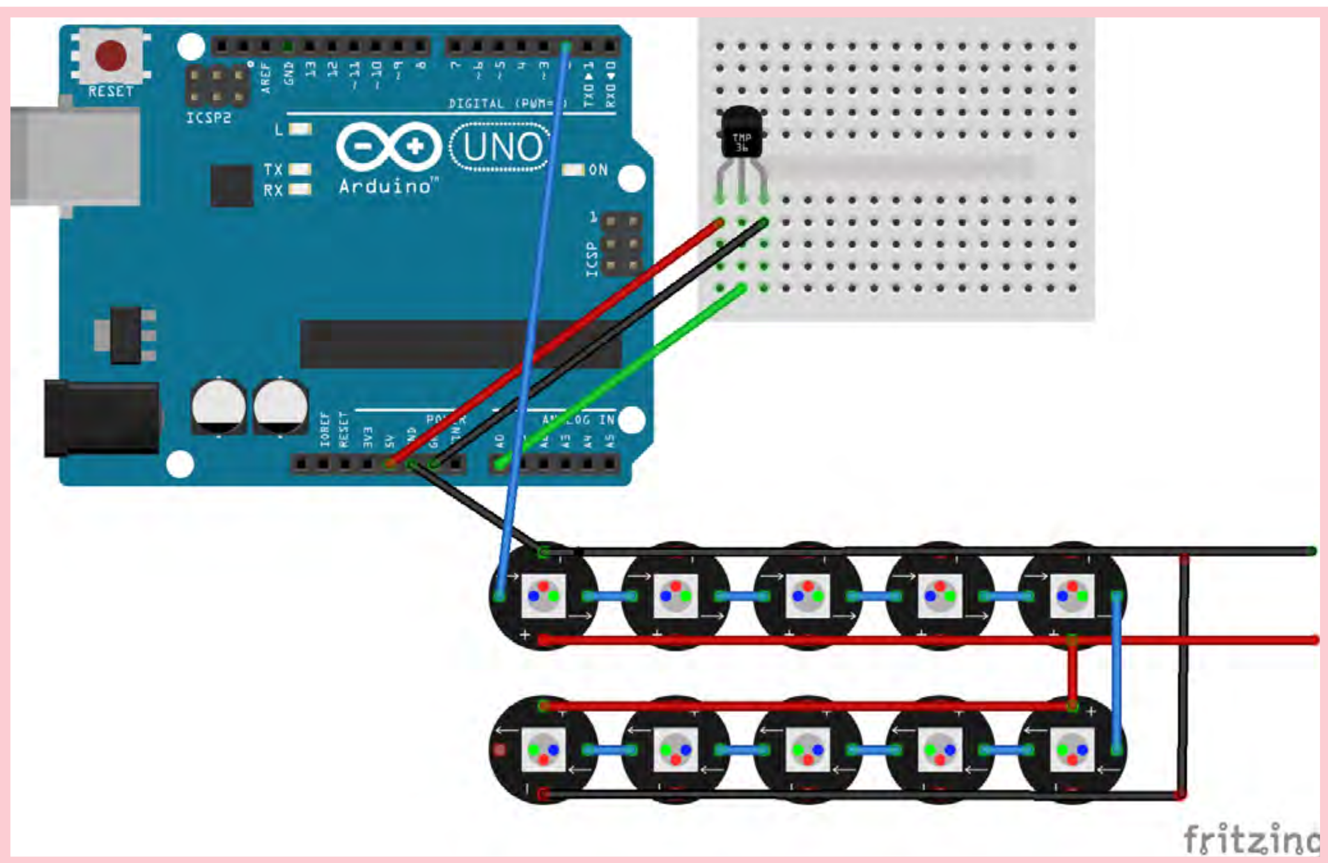
Power the Arduino Board with 6-10V using a USB cable connected to a computer.

3. Attach the input (middle leg of the sensor) to the analog pin A0 of the Arduino board (see drawing) through the breadboard.

4. Attach the output (middle pin of the LEDs) to digital pin 2 of the Arduino board with a soldered jumper wire (as for the + and -). Be careful: pin 2 should be connected to the first LED, not any LED !

For a full description of the wiring and power supply, please refer to the Adafruit Neopixel Uberguide (https://learn.adafruit.com/adafruit-neopixel-uberguide/powering-neopixels).





SENSORS, MICROCONTROLLERS AND ACTUATORS

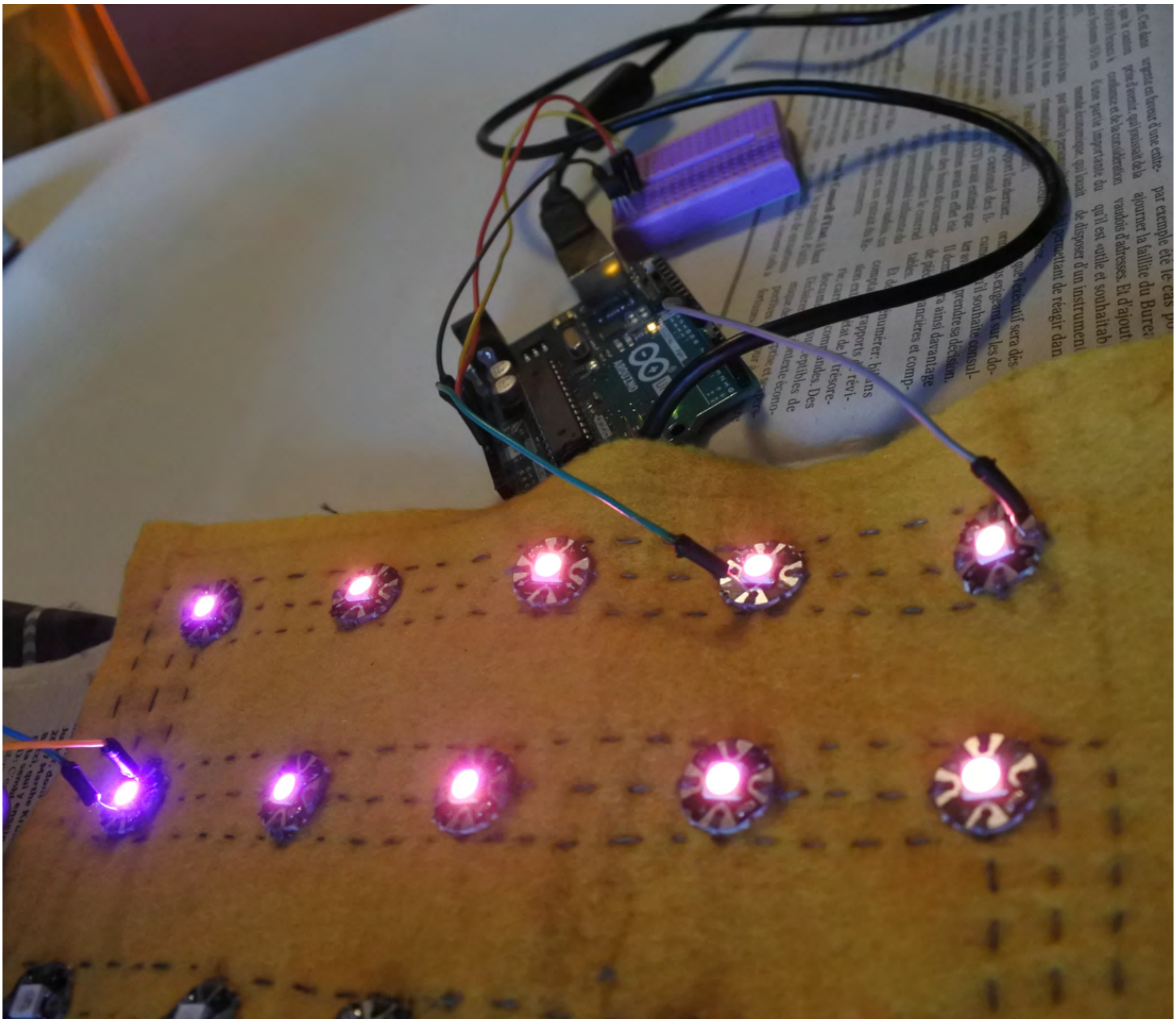
This crystal lamp that we are building is similar to a robot. Robots have 'senses' to detect information about their environment. In the lamp, the temperature sensor TMP36 can measure the temperature of the environment. Just like robots can process information, the Arduino UNO microcontroller in the lamp is coded to process data from the sensor according to our objectives. Finally, robots perform actions. The LEDs in this lamp changes colours based on the information provided by the sensor and processed by the microcontrollers.

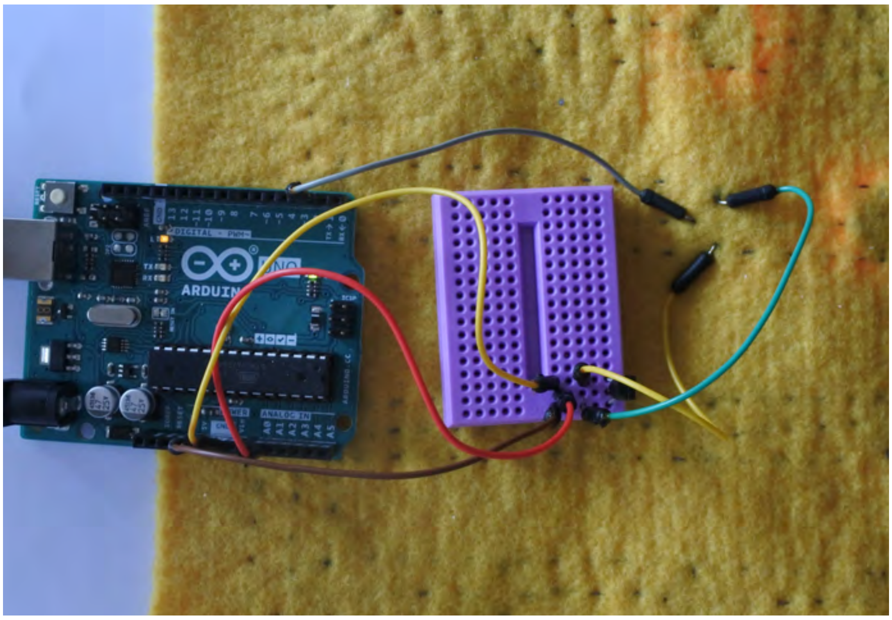
The main components are:

Sensor TMP36: Generates a voltage signal that is directly proportional to the temperature.

Light-emitting diodes (LEDs): When a suitable current is applied to the leads of the LEDs, the movement of the electrons within the device releases energy in the form of light (photons). In Neopixels, red, green and blue LEDs are integrated to produce different patterns of colours. Neopixels LEDs have a -, +, input and output pin.

Arduino UNO: Is an open-source microcontroller, a small computer, on a single integrated circuit; it is equipped with sets of digital and analog input/output (I/O) pins.





THE POWER SUPPLY

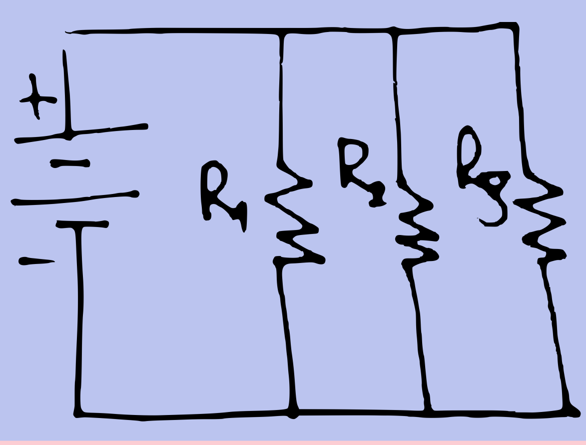
To correctly power the circuit that we have built, we need to make some simple calculations.

The Arduino Uno board can be powered either with a DC power jack (7 - 12V) or the USB connector (5V). The TMP36 sensor only needs 2.7V to 5.5V and has a 0.05 mA current draw. It can therefore be powered by the Arduino (3.3 or 5V).

Neopixels need 5V to 9V of DC to run, but each Neopixel draws as much as 60 mA (all three RGB LEDs on full brightness white). Our circuit has 10 Neopixels connected in parallel, which corresponds to a nominal total of 0.6 A. However, the nontrivial resistance of the conductive thread can affect the power supply. Because the current draw will add up fast, we recommend the use of a 5V, 2A power supply.

As long as the output is 5V of DC, you will not face any problems as the LEDs will only draw as much current (A) as they need. Excessive voltage, however, will damage your LEDs.

In a parallel circuit like ours, the voltage across each of the components is the same, and the total current is the sum of the currents through each component.



D.

THE

CODE

To program the Neopixels, first install Adafruit_NeoPixel via Library Manager. After having connected and powered the circuit, test the example codes contained in the library.
For a full description of how to install and connect an Arduino board, please refer to Getting Started with Arduino Uno (https://www.arduino.cc/en/Guide/ArduinoUno).
For a full description of how to install the Arduino Library, please refer to
Arduino Library Installation (https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-installation).

The code to control the LEDs is quite simple: depending on the temperature range, the LEDs will change colours.

neopixel

```
#ifdef __AVR__
  #include <avr/power.h>
#endif
#define PIN 2
#define NUM_LEDS 9
#define BRIGHTNESS 50

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUM_LEDS, PIN, NEO_GRB + NEO_KHZ800);

//TMP36 Pin Variables
int sensorPin = A0; //the analog pin the TMP36's Vout (sense) pin is connected to the resolution is 10 mV / degree centigrade with a 500 mV

int neopixel = 2;

```

In the variable section, we defined the output pin for the LEDs (2), the number of LEDs (10), the brightness (50), the input pin for the sensor (A0) and called the libraries.
The Void Setup section, which runs when the program begins, lays the foundation for the actions executed later in the program. The input and output are defined here.

```
void setup()
{
  pixels.setBrightness(BRIGHTNESS);
  pixels.begin();
  pixels.show(); // Initialize all pixels to 'off'
  pinMode (neopixel, OUTPUT); // the neopixel pin will be used to control an output device
  pinMode (sensorPin, INPUT); // the TMP36 sensor tells the Arduino the kind of sensor
  Serial.begin(9600);  //Start the serial connection with the computer
                       //to view the result open the serial monitor
}
```

After the setup section runs, the loop section runs over and over until the Arduino is turned off. The first part of the code is for the temperature sensor. You can read the temperature in degrees Celsius on the Arduino serial monitor to test if it works fine.

```
void loop()                    // run over and over again
{
//getting the voltage reading from the temperature sensor
 int reading = analogRead(sensorPin);

 // converting that reading to voltage, for 3.3v arduino use 3.3
 float voltage = reading * 5.0;
 voltage /= 1024.0;

  // print out the voltage
 Serial.print(voltage); Serial.println(" volts");

 // now print out the temperature
 float temperatureC = (voltage - 0.5) * 100 ;  //converting from 10 mv per degree wit 500 mV offset
                                               // to degrees ((voltage - 500mV) times 100)
 Serial.print(temperatureC); Serial.println(" degrees C");



 delay(1000);
//waiting a second
```

The second part of the code is also simple: it tells the LEDs to change colour for every variation of 0.5 degree Celsius. You can choose the colours you prefer and set bigger or smaller temperature fractions. Remember to close the code with a curly bracket.

```
or(int i=0;i<NUM_LEDS;i++) {
//https://htmlcolorcodes.com/color-names/

f (temperatureC > 10 && temperatureC < 10.5  ) {


   // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
   pixels.setPixelColor(i, pixels.Color(25,25,112));

   pixels.show(); // This sends the updated pixel color to the hardware.

 }
 if (temperatureC > 10.5 && temperatureC < 11 ) {


   // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
   pixels.setPixelColor(i, pixels.Color(0,0,205)); //

   pixels.show(); // This sends the updated pixel color to the hardware.

 }
 if (temperatureC > 11 && temperatureC < 11.5  ) {


   // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
   pixels.setPixelColor(i, pixels.Color(123, 104, 238));

   pixels.show(); // This sends the updated pixel color to the hardware.

 }
 if (temperatureC > 11.5 && temperatureC < 12  ) {


   // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
   pixels.setPixelColor(i, pixels.Color(0,191,255)); //

   pixels.show(); // This sends the updated pixel color to the hardware.

 }
```

......

```
   // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
   pixels.setPixelColor(i, pixels.Color(178,34,34));

   pixels.show(); // This sends the updated pixel color to the hardware.

 }
 if (temperatureC > 32 && temperatureC < 32.5 ) {


   // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
   pixels.setPixelColor(i, pixels.Color(128,0,0));

   pixels.show(); // This sends the updated pixel color to the hardware.


 }
```

Test your circuit and code before moving to the next step.

CODING

Computers like Arduino UNO execute orders – a program – to perform actions. We need to learn the computer's language. Just like human languages, the computer language has many idioms and words.

According to L. Buechley's structure of an Arduino program, each code has 3 main sections:

1. Variable declaration
2. Setup
3. Loop

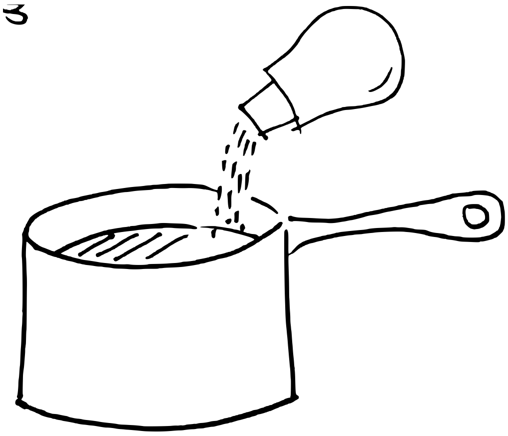
These sections are analogous to the ingredient list (variable declaration), preparation (setup) and the cooking process (loop) in a recipe. When your program runs, it will first define your variables (the ingredients that you need), then execute the setup once (arrange everything to begin cooking) and then execute the loop over and over (actually do the cooking).

E.

THE CRYSTALS

1. Coat the conductive threads and LEDs with transparent nail polish to prevent the connections from being corroded by the crystals. Coat both sides of the felt. .

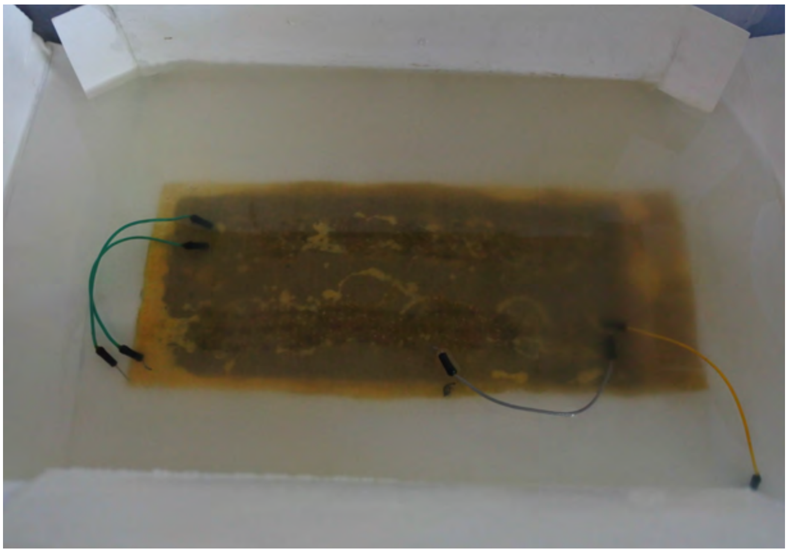
3

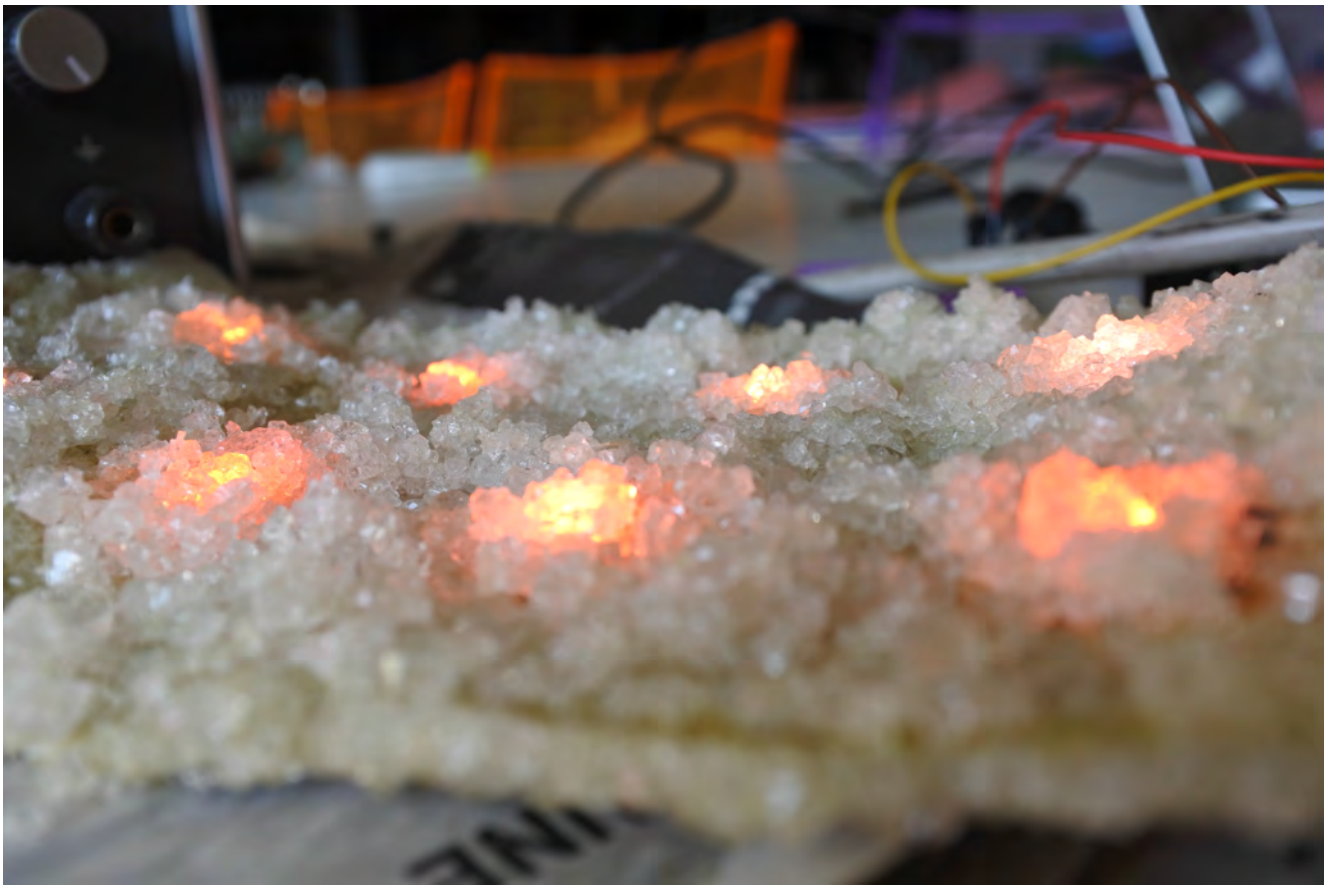


2. Dissolve 700 g of alum in 1 litre of hot water. Once the alum is completely dissolved and the solution is supersaturated, pour it into a plastic basin.

3. Immerse the felt with the circuit into the solution with the LED side facing down. Do not move the basin for at least 24 hours.

4. After 1-3 days, remove the felt and let it dry: Eventually, the crystals will grow and be strongly bound to the LED layer. The felt will be covered by a layer of 2-3 mm of crystals.





CRYSTALS

A crystal is a solid object with symmetrically arranged flat surfaces that meet in straight lines and sharp corners. At the microscopic level, the molecules of a crystal are arranged in regular repeating patterns. Salts, diamonds or snowflakes are known examples of crystals. All crystals grow from a seed, a small single crystal to which more material gets attached. To grow crystals, we need saturated (no more substance can dissolve in it) solutions of water (or other solvents) that should be allowed to evaporate slowly and remain undisturbed. One of the larger crystals is selected to be the seed crystal and immersed in a new saturated solution. The new solution must be cool, or it will dissolve the seed crystal.

Alum is the name of a group of hydrated double salts, usually consisting of aluminium sulphate, water of hydration and the sulphate of another element. Alums are usually colourless, odourless and exist in white crystalline powder form. The can be easily produced by precipitation from an aqueous solution. To produce potassium alum, for example, aluminium sulphate and potassium sulphate are dissolved in water, and upon evaporation, alum crystals are formed from the solution. A whole series of hydrated double salts results from the hydration of the sulphate of a singly charged cation (e.g., K^+) and the sulphate of any one of a number of triply charged cations (e.g., Al^{3+}). Potassium aluminium sulphate has the the following molecular formula:

$K_2(SO_4)\cdot Al_2(SO_4)_3\cdot 24H_2O$ or $KAl(SO_4)_2\cdot 12H_2O$.

F.

THE STRUCTURE

1. Once the structure is completely covered with crystals, proceed with the final attachments.

2. In the centre of the paper frame, cut out a rectangle of 18 cm x 8 cm and stick the felt to the frame with a strong glue.

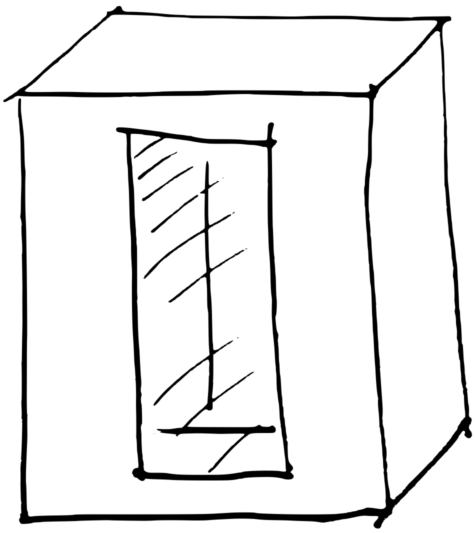
3. With double-sided tape, fix the Arduino and the breadboard with the temperature sensor to the interior of the box, as shown below.

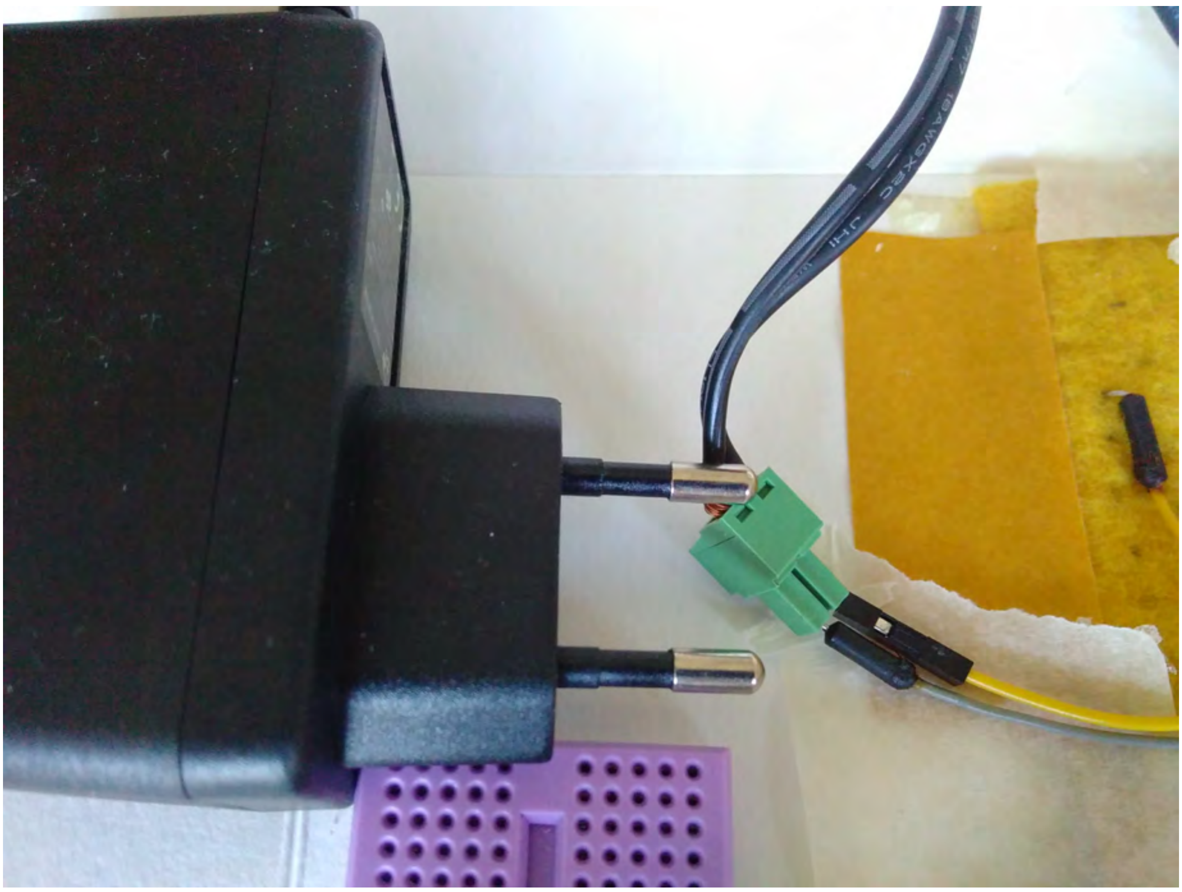
4. Gently close the box and if needed tape it to keep it closed.

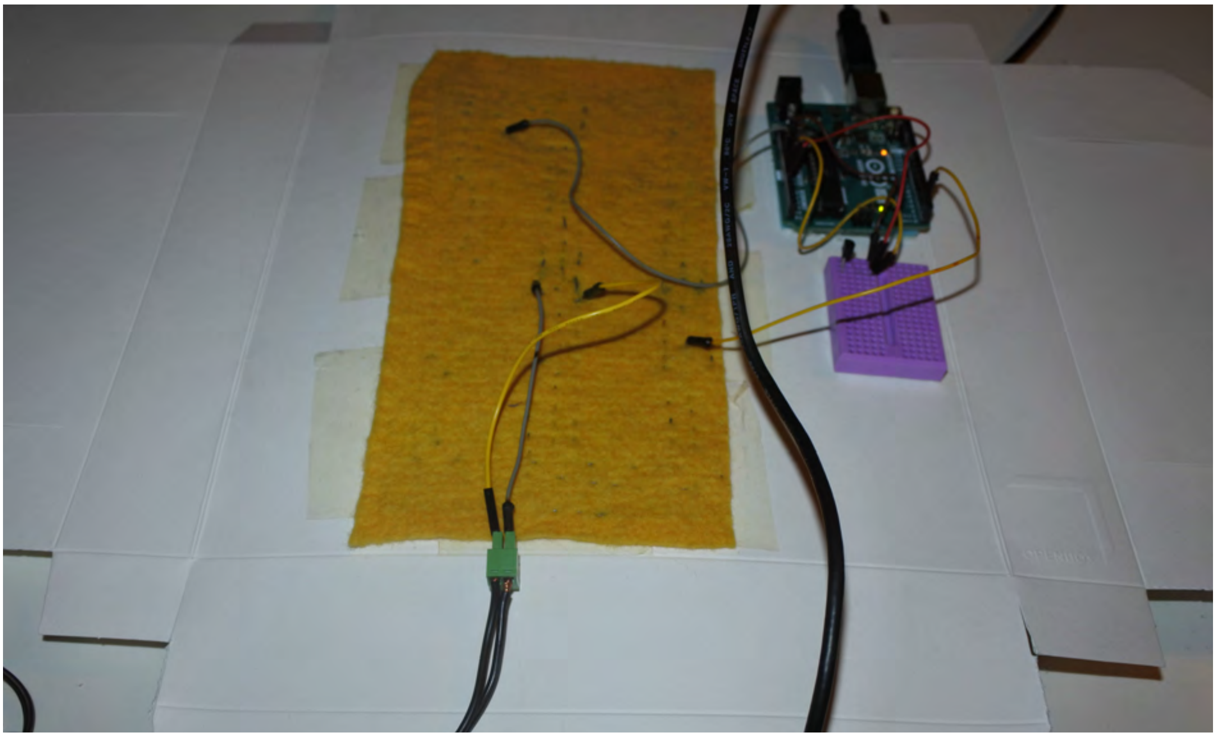
5. Switch on the power supply of the LEDs first and then of the Arduino.

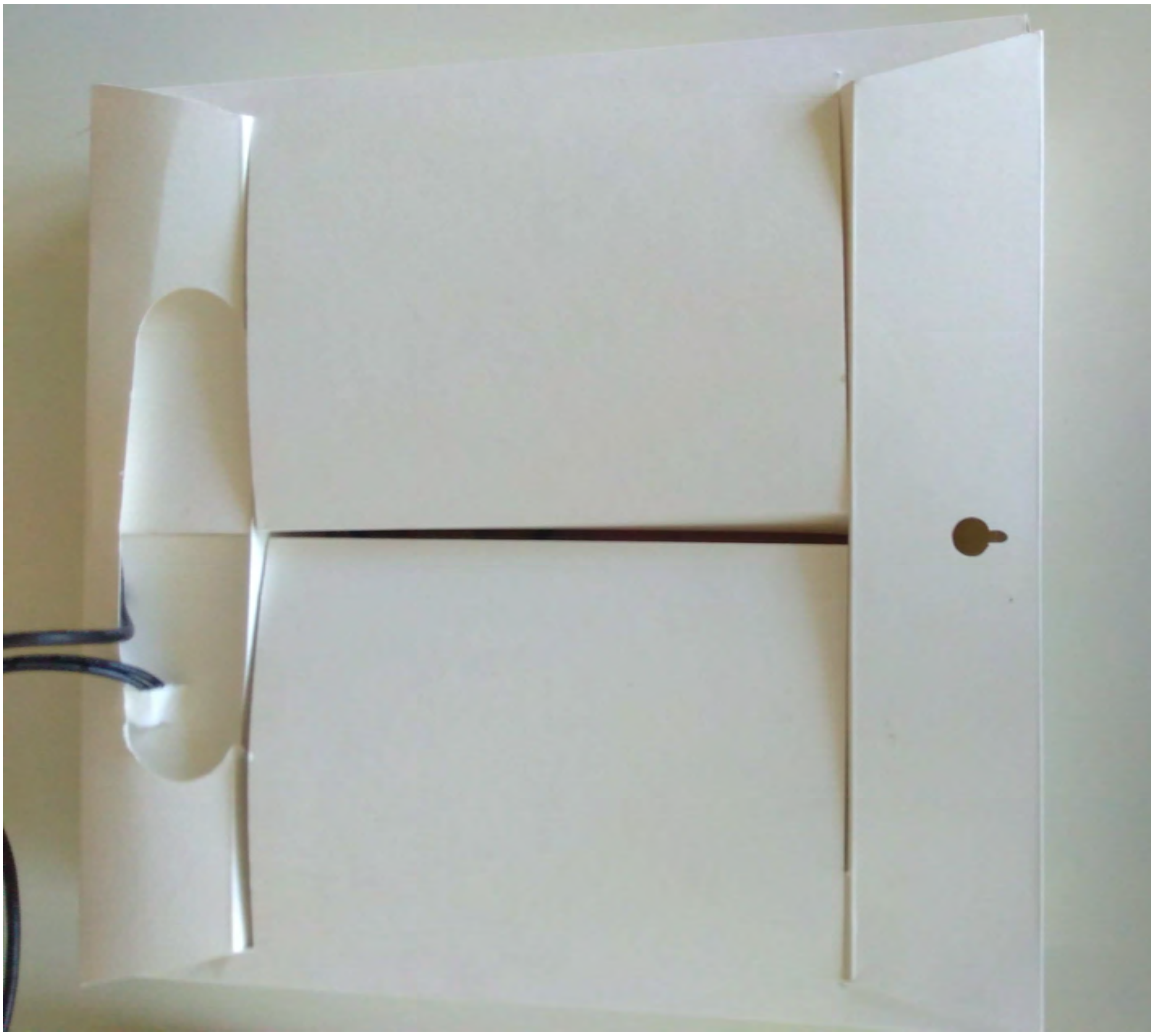
6. Power the Arduino by plugging in the power supply or the USB cable to the computer.

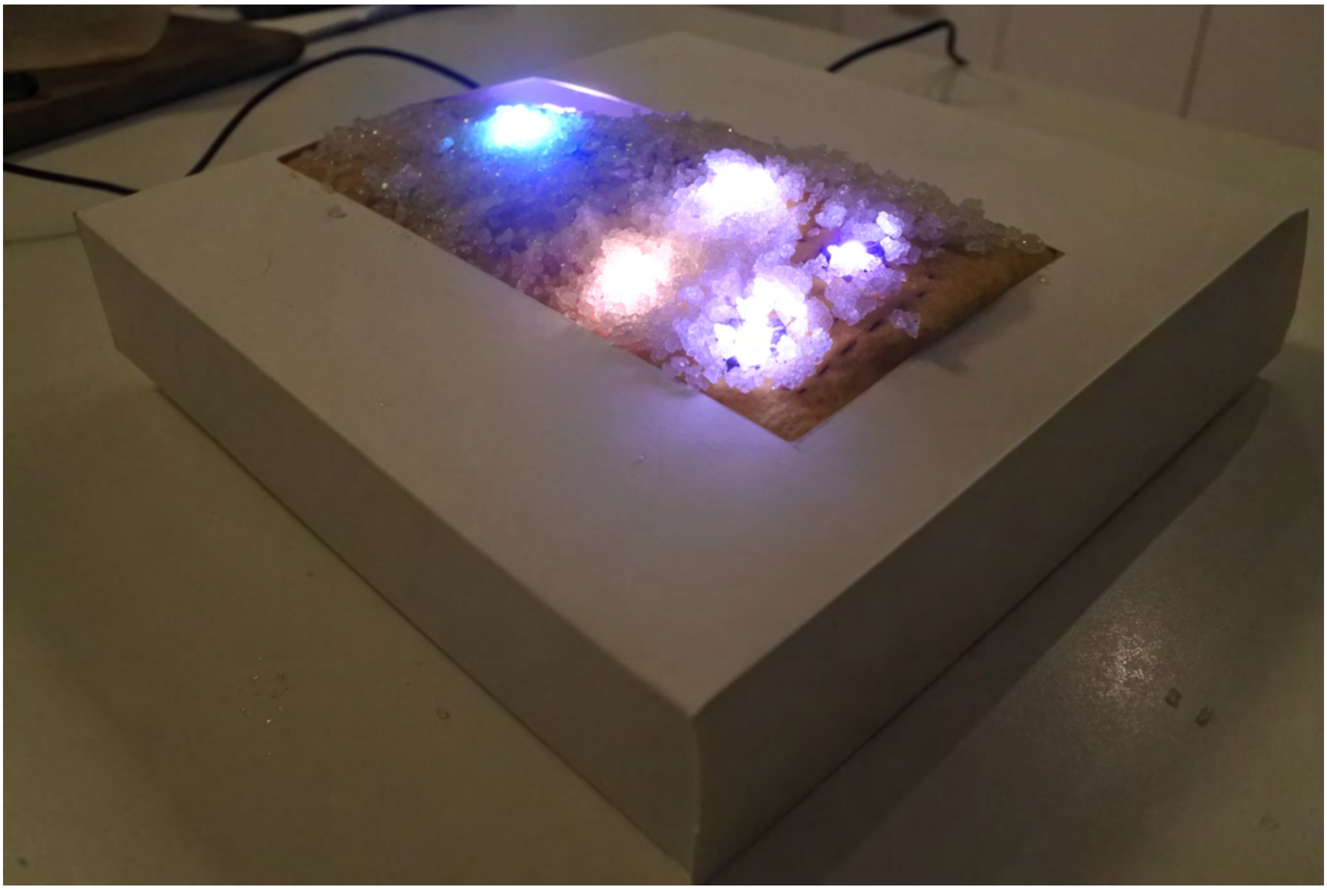
Your lamp is ready!

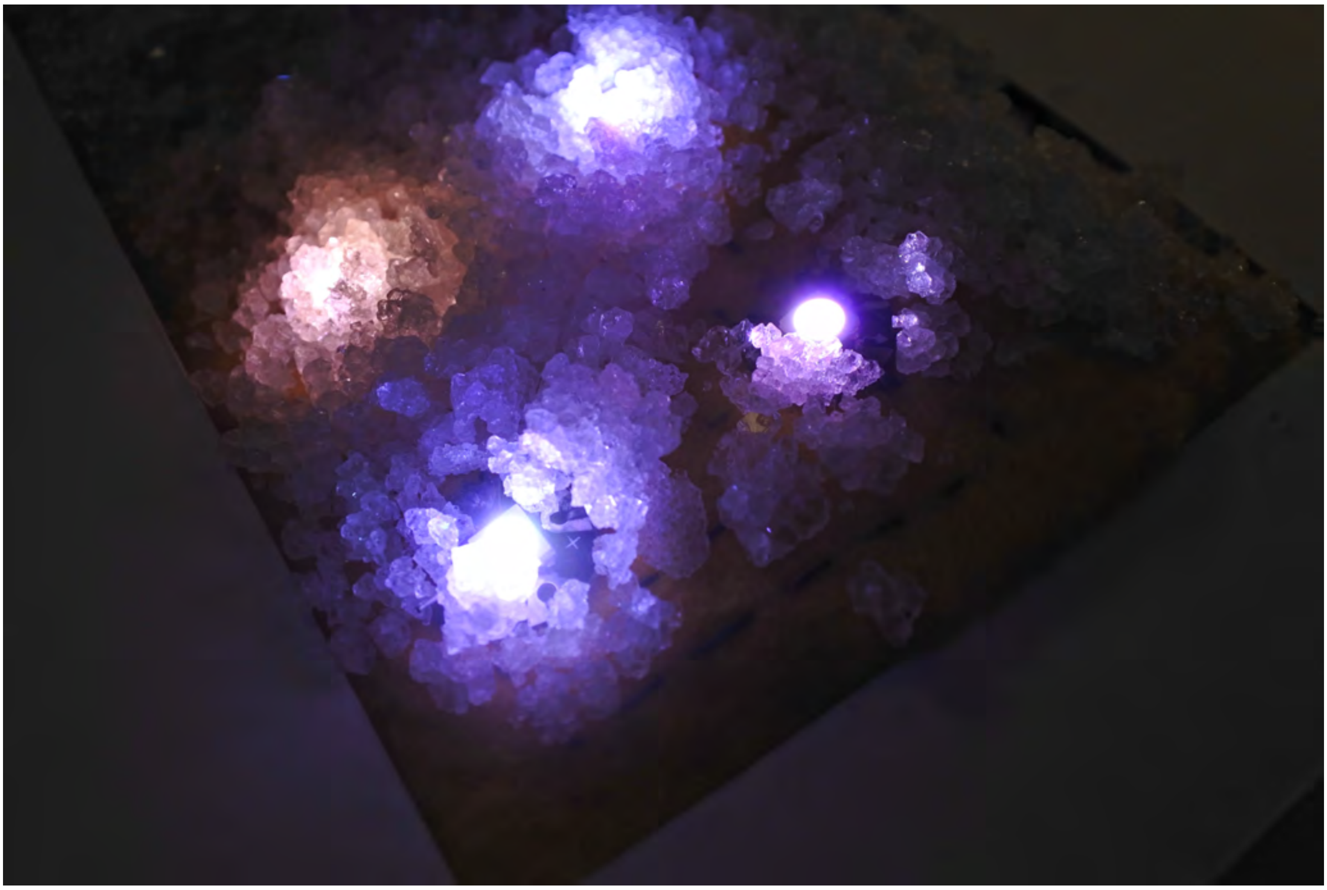












G.

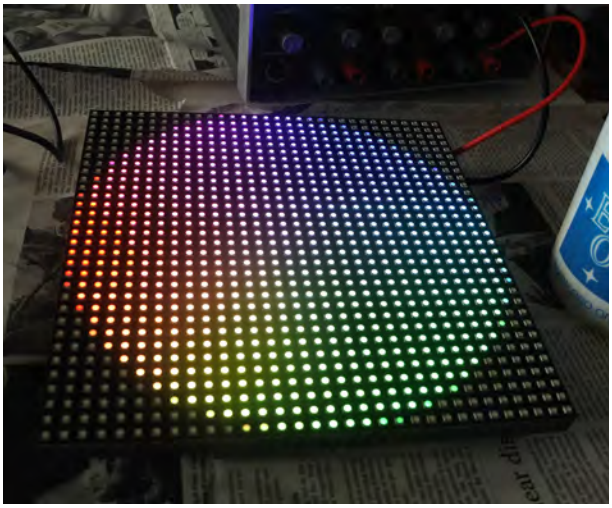
ELABORATION

1. Instead of sewing an array of LEDs, you can opt for a 32 x 16 or 32 x 32 RGB LED Matrix. The matrix will replace:

- Thin lightly coloured felt
- Vinyl glue
- Conductive thread
- Needles
- Neopixel LEDs
- Solder
- Soldering irons

2. Instead of a temperature sensor, you could use a different sensor to measure sound levels or luminosity. The structure of the code will be the same, but you will need to learn the characteristics of the sensors for connections and supplies.

3. The paper frame we used was obtained from a Crowdfunding project called OpenFrame by Filippo Protasoni and Ciro Trezzi. You can also design your own paper frame, which can be laser-cut or cut manually.



G.

EVALUATION

The evaluation covers three different aspects of the project:

- Work methodology (observation)
- End product (observation)
- Science concepts (oral and/or written)

Electronics	Sensors	Did the student understand what is a sensor and how it should be connected to a micro controller?	- -	-	+	+ +
	Power supply	Is the student able to calculate the power supply needed by the circuit? Does he/she understand the concept of ground? Of short circuit?				
	Microcontroller	Is the student able to identify the main features and pins of the microcontroller?				
IT	Coding	Is the student able to understand the structure of an Arduino code? Is she/he able to modify variables? Is he/she able to understand the logic 'if, then...'?				
Arts	Fine motor skills	Is the student able to work neatly and use the needed tools appropriately?				
Science	Circuits	Did the student understand the concept of an electric circuit? The difference between circuits in parallel and series? The relation between voltage, intensity, resistance and power?				
	Crystals	Did the student understand the conditions needed for growing crystals? Is he/she able to define what is a crystal and describe its microscopic structure?				

Social	Team work	Did the student work actively in the team? Did he/she help other teams if needed?				
	Problem solving	Did the student try to solve problems by him/herself before asking for help from the teacher?				
	Creativity	Did the student propose variations/improvements to the given instructions?				
	Autonomy	Was the student able to plan his/her time and progress accordingly?				

—

QUESTIONS?
COMMENTS?
SUGGESTIONS?
CONTACT@ONLFAIT.CH