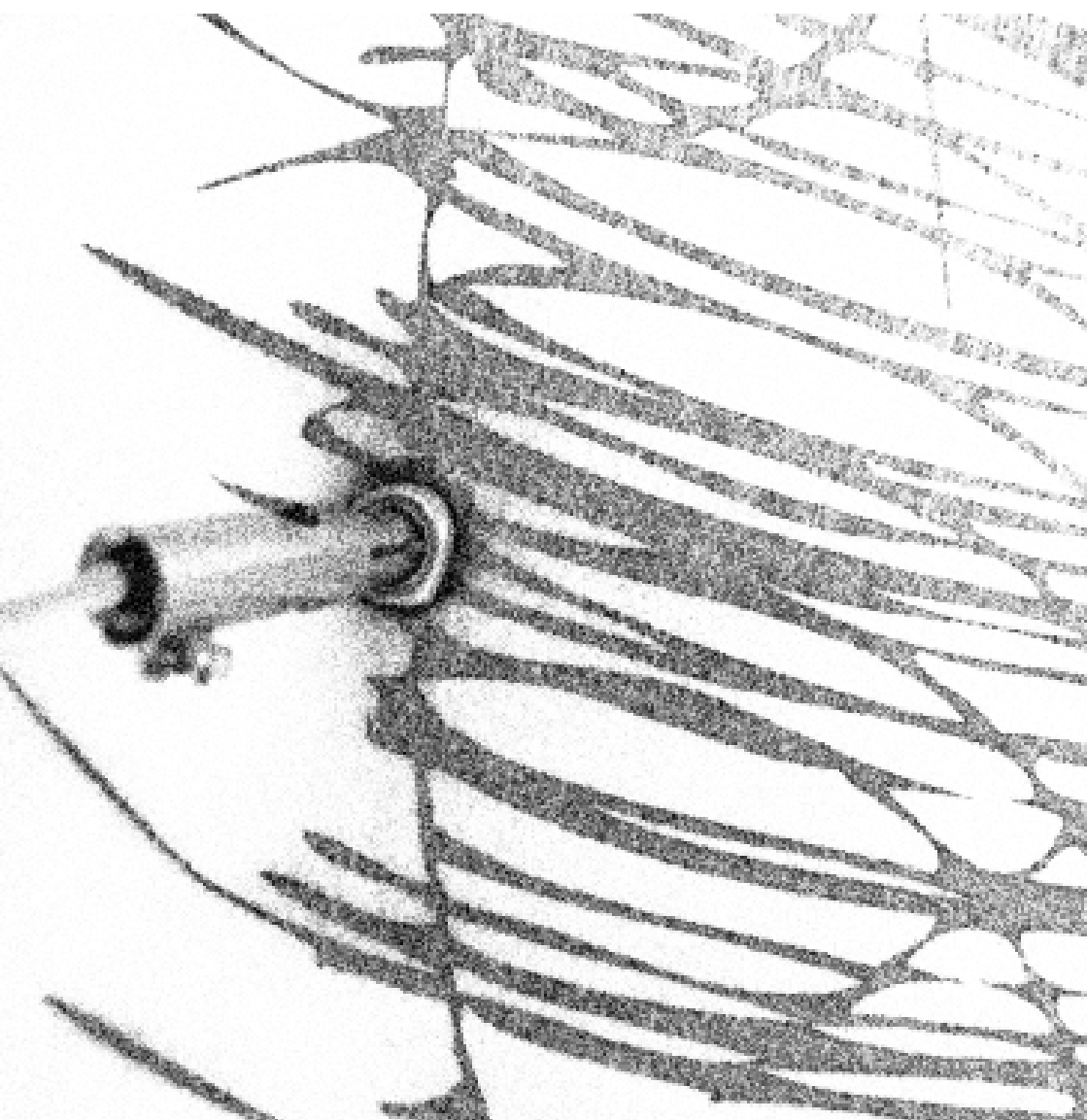
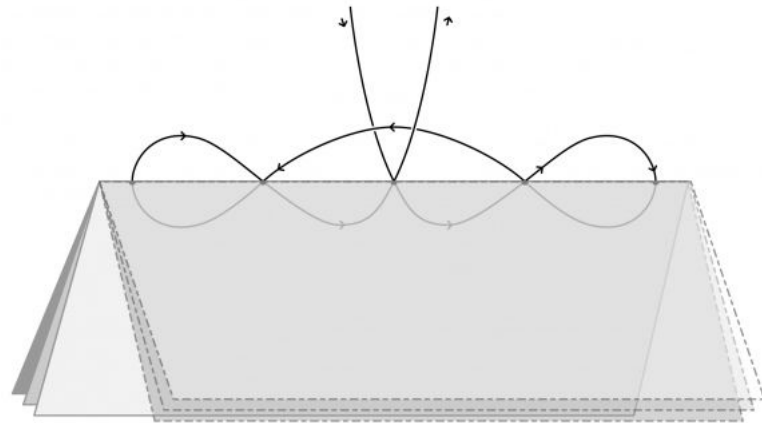


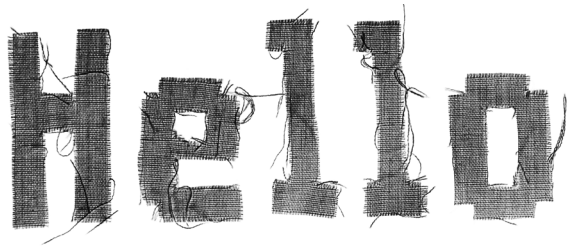
E-TEXTILE READER TECHNOLOGY



CONTENTS

Hello and welcome	4
Meet the actors	
Materials.....	10
Electrical components.....	12
What is electricity	
Electrical charge.....	14
Ohm's law.....	16
Multimeter.....	17
Build your first circuit:closed and switch.....	18
Build your second circuit:variable resistance...	20
Linking real world and computer	
Arduino - what is this??.....	22
Hello World.....	23
DigitalIn: on/off button.....	24
AnalogReadSerial: variable resistance.....	28
Real world data to computer signals.....	32
Humans and computers interact	
Graphical programming.....	34
Pure Data basics.....	36
Hello World (440Hz).....	38
Play sound file.....	40
GEM (Graphics Environment for Multimedia.....	42
Serial read: Pduino.....	44
Mapping sensor data.....	46
Resources	
Materials and components.....	48
Learning & tutorials on the web.....	50
Literature theory / context.....	51
References and imprint	52





Welcome to the **Crafting Textile** **Data reader.**

We made this zine to guide you through the technical aspects of the seminar. It follows the sessions and exercises chronologically, but you can read it in any direction and as often as you like.

This is the first time we are teaching this seminar and in advance we thought quite a lot about what is important for us as designers and teachers working with e-textiles. So we decided to make this short intro about us and describe a few things we believe in. It may be useful for you as background information to contextualise the journey we intend to take you on.

We are **Berit** and **Giorgia**.



I am Berit. Since August 2018 I work as a junior professor for Wearable Computing at UdK at the Institute for Experimental Fashion and

Textile Design. I am passionate about materials, and discovering the beauty of science through artistic inquiry. My background is in textile design. But I believe in pushing disciplinary boundaries, and after studying textiles at KHB-Weissensee and at Central St Martins I went on to do a PhD in electromagnetic engineering.



My name is Giorgia, I am a PhD student in Wearable Computing at the University of the Arts in Berlin. I worked in Milan in the field of

wearable and e-textile and as a freelance interaction developer and A/V artist. My work links art and science with a particular interest in nature and its connection with technology. My background is in aerospace engineering and new technologies for art, and that's why I like to mix art and science.

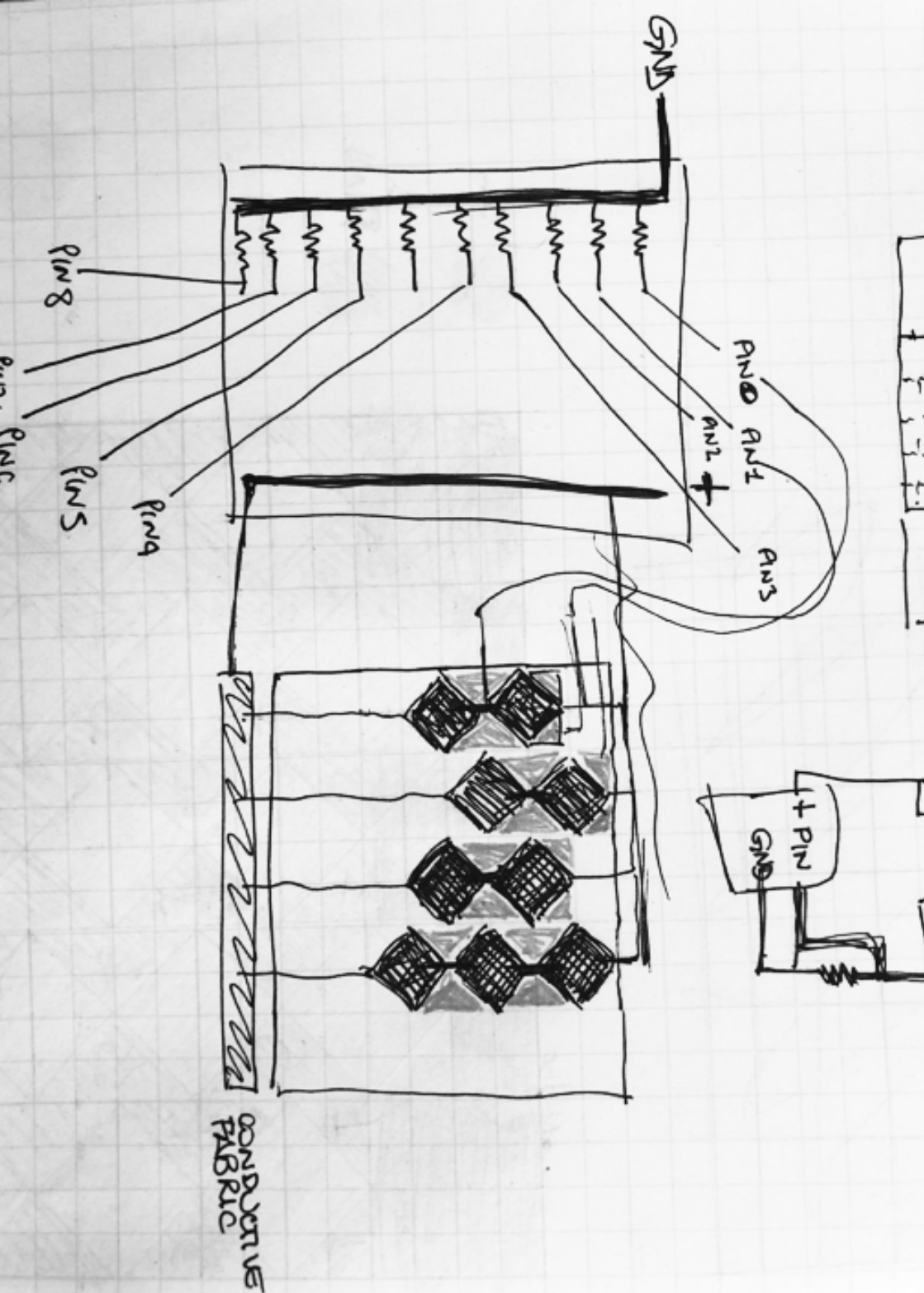
We believe in:

Making "Almost all of us can make. It is one of the strongest of human impulses and one of the most significant means of human expression. To some, making is the fountain that releases creative ideas; to others, making is about participating in society as well as defining personal identity. [...] While for some people making is crucial for survival, for others it is a way of learning. And maybe also a way of defying conventions, enjoying life or solving its problems. Making serves other needs, too. It allows people to take care of loved ones, worship, mourn, celebrate or demonstrate. It is a way of exercising (free) will."
(A quote from Daniel Charny, who was a guest curator for the exhibition 'The Power of Making' at the Victoria and Albert Museum in 2011.)

Open Source The e-textile crafts community advocates the use of Open Source technology and open knowledge sharing. The term Open Source originated in the software community in the 1980s, and promotes free and transparent redistribution, fairness and changeability of source codes. You can read more about the 10 criteria on the website of the Open Source Initiative on <http://www.opensource.org/osd>. Curiously, there is much overlap between open source and thinking in some historical crafts movements, aiming for social inclusion and equality. In broad terms, we advocate for transparency, integrity, and inclusiveness - no matter if it concerns source codes, textile crafting or garment design.

Kindness We're sure you know these, but here they are again:

Learn from and with others. Help each other. Let yourself be surprised. Be kind. Be curious. Tell jokes.

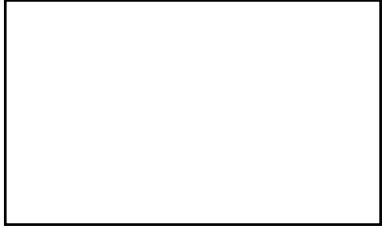


We recommend:

1. Bring a sketchbook to class, and take as many notes as you can. We believe everyone's learning style is and should be different, and notes can be anything from drawings, text, images, material samples, recordings etc. Use a format that is easily accessible when going back to it after longer periods of time, and helps you understanding and memorising, rather than causing more work.
2. You might have heard this before but as it is so important we say it again: there are no stupid questions. Please ask us anything. You can ask us in class, after class, during open office hours or per email - whatever you feel most comfortable with. (Bonus for us: Your questions also help us to learn from you. What are your interests, questions, struggles, critiques, and how can we adapt our teaching to your learning styles?)
3. There is a lively e-textile community out there, which loves to make, share, exhibit, write, perform and teach. They are artists, designers, scientists, engineers and general humans who are fun, critical and curious disciplinary-crossers. Keep an eye out for opportunities to meet them! We're always happy to let you know about upcoming events - and hear about others from you.

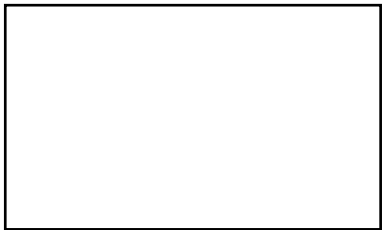
MEET THE MATERIALS

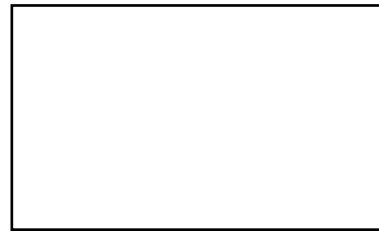
E-TEXTILES

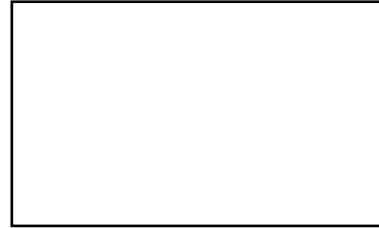




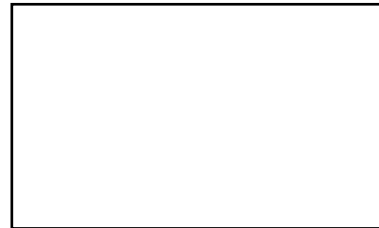


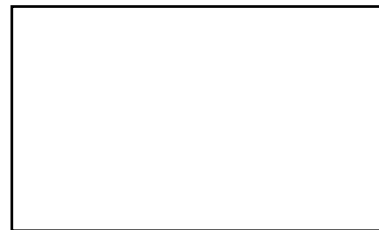






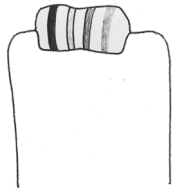






MEET THE COMPONENTS

ESSENTIALS



Resistor

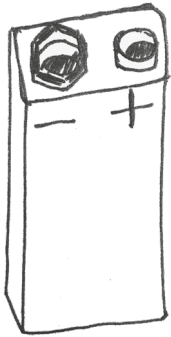
Resistors are electronic components which have a specific, never-changing electrical resistance. The resistor's resistance limits the flow of electrons through a circuit. Resistors - the most ubiquitous of electronic components. They are a critical piece in just about every circuit.



Battery / Power

Batteries are a collection of one or more cells whose chemical reactions create a flow of electrons in a circuit. All batteries are made up of three basic components: an anode (the '-' side), a cathode (the '+' side), and some kind of electrolyte (a substance that chemically reacts with the anode and cathode).

When the anode and cathode of a battery is connected to a circuit, a chemical reaction takes place between the anode and the electrolyte. This reaction causes electrons to flow through the circuit and back into the cathode where another chemical reaction takes place. When the material in the cathode or anode is consumed or no longer able to be used in the reaction, the battery is unable to produce electricity. At that point, your battery is "dead."



Crocodile lead

A crocodile clip (also alligator clip) is a sprung metal clip with long, serrated jaws which is used for creating a temporary electrical connection. It gets its name from the resemblance of its jaws to that of an alligator's or crocodile's. It is very useful when working with e-textiles, as textile components do not have standardised connection points. A crocodile clip can be easily clipped to a fabric.

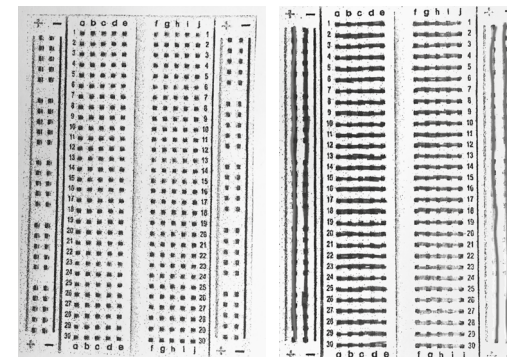
Jumper wire



Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards (see next page) and other prototyping tools in order to make it easy to change a circuit as needed.

Breadboard

Great having all these components, but how to stick them together? That's what a breadboard is for - simple and quick prototyping to test if your circuit works. Breadboards have a standardised layout of connections (see below right).



THIS IS
WHAT YOU SEE

THIS IS
WHAT
YOU GET

MEET THE COMPONENTS

OUTPUT / ACTUATORS

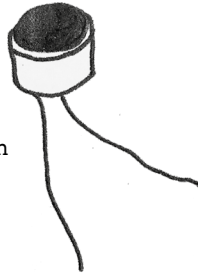


Light-emitting-diode (LED)

LEDs (that's "ell-ee-dees") are a particular type of diode that convert electrical energy into light. In fact, LED stands for "Light Emitting Diode." (It does what it says on the tin!) In short, LEDs are like tiny lightbulbs.

Buzzer

A buzzer (or beeper) is an audio signalling device. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.



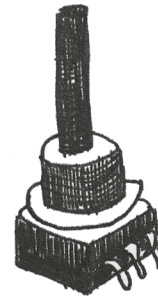
INPUT / BINARY SENSOR



Switch (or Button)

A switch in an electronic device is used to interrupt the flow of electricity or electric current. Electrical switches are binary devices, they can be either completely off or completely on.

INPUT / VARIABLE SENSORS

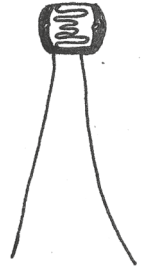


Potentiometer

Potentiometers (also known as "pots" or "knobs") are one of the basic inputs for electronics devices. A potentiometer is a variable resistor. By tracking the position of the knob, you can make volume controls, speed controls, and a ton of other useful inputs for your projects.

Light Sensor

A photoresistor (or Light Dependant Resistor, short: LDR) is a component that has a (variable) resistance that changes with the light intensity that falls upon it.



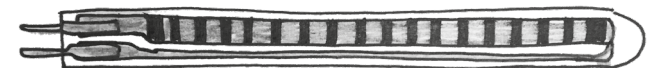
Pressure Sensor

A pressure sensor, as the name suggests, is an electronic component that senses and measures mechanical pressure. Resistance usually decreases when it is pressed.



Flex Sensor

A flex sensor or bend sensor is a sensor that measures the amount of deflection or bending.



ELECTRICAL CHARGE VOLTAGE, CURRENT, RESISTANCE

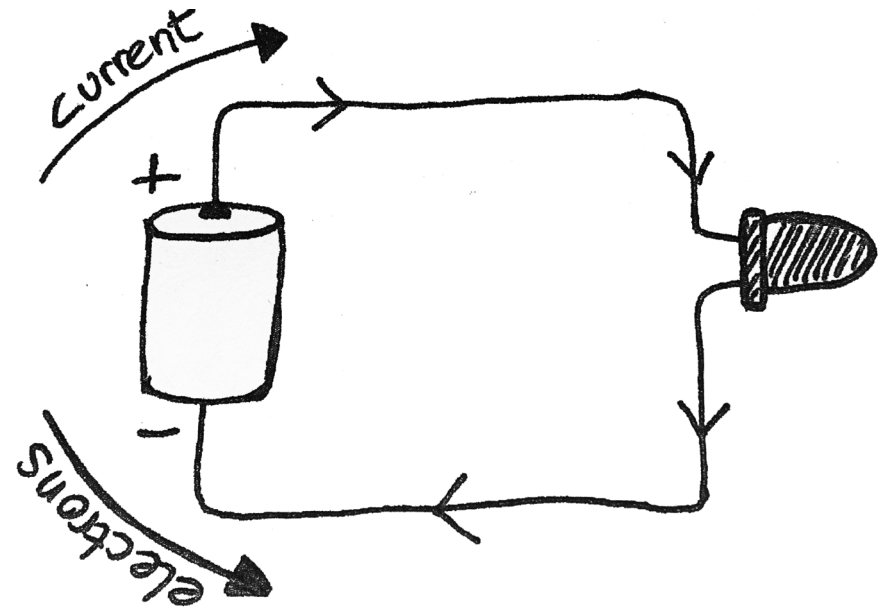
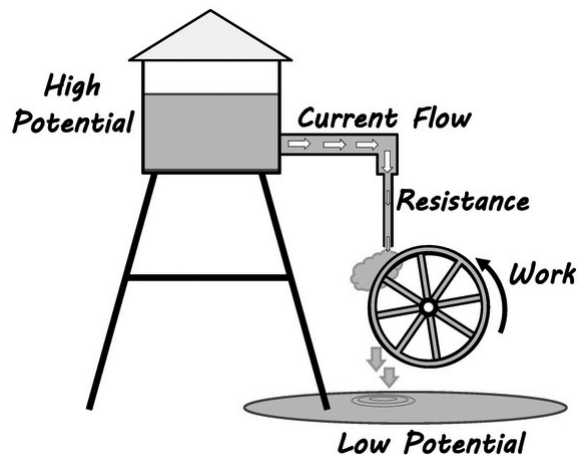
Electricity is the movement of electrons. Electrons create charge, which we can harness to do work. Your lightbulb, your stereo, your phone, etc., are all harnessing the movement of the electrons in order to do work. They all operate using the same basic power source: the movement of electrons.

The three basic principles can be explained using electrons (see page on the right). A good analogy is water and a water pump, see below right.

Voltage is the difference in charge between two points (potential).

Current is the rate at which charge is flowing (flow).

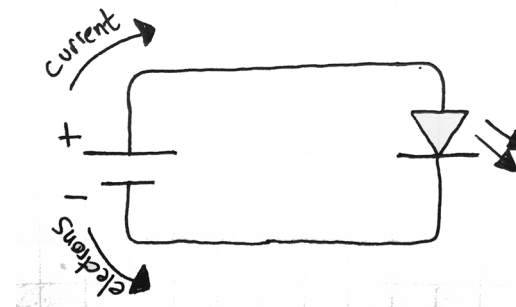
Resistance is a material's tendency to resist the flow of charge.



When we talk about these values, we're really describing the movement of charge, and thus, the behaviour of electrons. A circuit is a closed loop that allows charge to move from one place to another. Components in the circuit allow us to control this charge and use it to do work for us.

The way that circuits are drawn in technical fields is standardised. So if you need to communicate with people from electronic engineering or manufacturing, this is the way to go.

But we encourage you to invent your own signs and language for these if you prefer - for now you are the only person who needs to understand and remember.



OHM'S LAW

Extending what you learned on the previous page:

Voltage (V) is the difference in charge between two points - **measured in VOLT (V)**. Named after



Alessandro Volta
1745 - 1827

Current (I) is the rate at which charge is flowing - **measured in Ampere (A)**. Named after

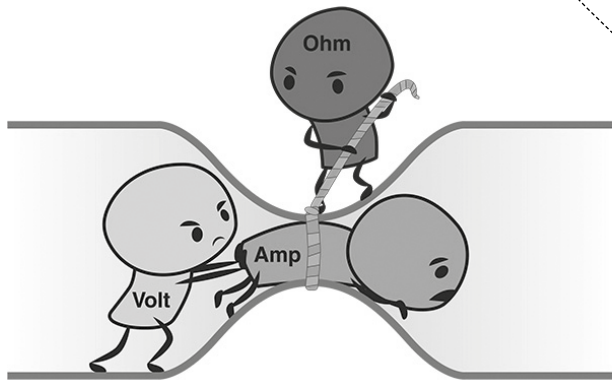


André-Marie Ampère
1775 - 1836

Resistance (R) is a material's tendency to resist the flow of charge (I) - **measured in Ohm (Ω)**. Named after



Georg Simon Ohm
1789 - 1854



Combining the elements of voltage, current, and resistance, Ohm developed the formula:

$$V = I \times R$$

EXAMPLE WHAT
IT IS GOOD FOR
PLEASE??

For example, it's good
for calculating what
size of resistor you
should add to your
circuit to avoid blowing
your LED.

MULTIMETER AND HOW TO USE IT

A **Multimeter** is an indispensable tool that you can use to diagnose circuits, learn about other people's electronic designs, and even test a battery. Hence the 'multi'-'meter' (multiple measurement) name.



Got an old battery lying around and want to know how much power is still left in it? Use your multimeter to find out!

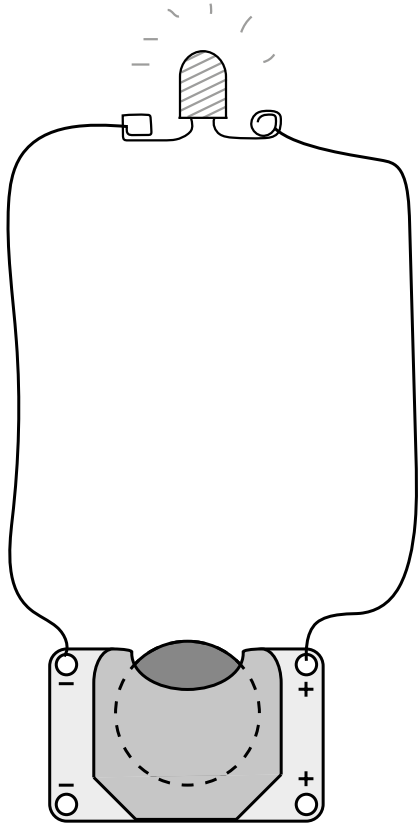
Measure voltage

Shopping at Hüco, and want to know if a fabric contains conductive metal? Bring your multimeter!

Measure resistance

BUILD YOUR FIRST CIRCUIT!

CLOSED AND SWITCH



You need:

- Coin cell CR2032
- Coin cell holder
- LED
- Conductive thread or fabric
- Non-conductive fabric

Assemble the circuit as illustrated. Remember that LEDs have a polarity, so you need to attach the LED the correct way round.

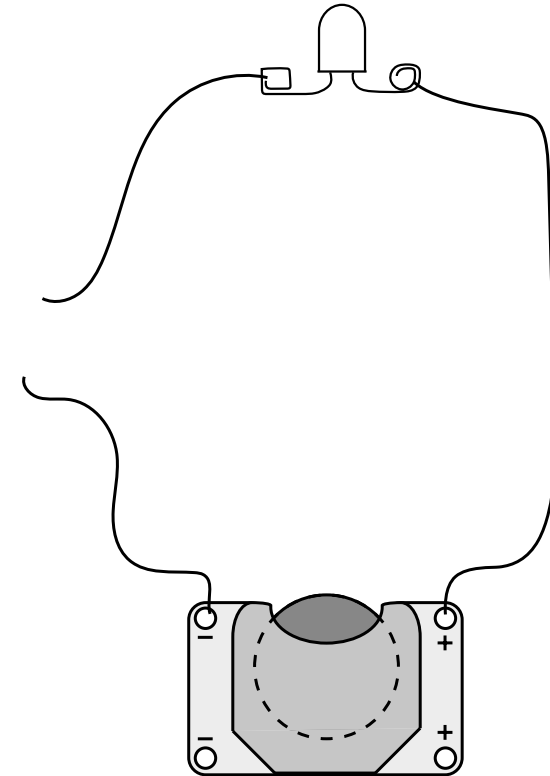
long leg = +
short leg = -

Use conductive thread or conductive fabric for the connections. Use non-conductive fabric as the carrying material.

Does your LED receive the right amount of voltage? Use a multimeter and Ohm's law to find out, and adjust the resistance of your conductive connection.

Now make a second circuit, but use a mechanical "switch" which allows you to interrupt the flow of electrons (LED off), and reconnect it (LED on).

Switches can be made in all shapes and sizes, and can be inserted at every point in the circuit. Think of how you could use your body or movement to trigger it.



A few examples for inspiration:

Fabric Button <https://www.kobakant.at/DIY/?p=48>

Crochet Button <https://www.kobakant.at/DIY/?p=1927>

Zipper Switch <https://www.kobakant.at/DIY/?p=2620>

Zebra fabric stroke sensor: <https://www.kobakant.at/DIY/?p=6163>



BUILD YOUR SECOND CIRCUIT!

VARIABLE RESISTANCE

Instead of just switching the LED on and off with a binary switch, you can use a variable sensor to change the brightness of the LED. You have already learned about variable sensors as electrical components (see page 11), but you can make these also from textile materials!

Build a pressure sensor, and use it in your circuit instead of a binary switch.

You need:

- Conductive fabric
- Pressure sensitive material (Velostat)
- Non-conductive fabric
- Iron-on fabric ("Hexenspucke")
- Non-conductive thread

Cut the paper patterns as provided. Then follow instruction here.

There are many examples of other textile variable sensors, such as pressure, slide, or flex sensors. Make a second variable resistor, but use a different material, type or textile technique.

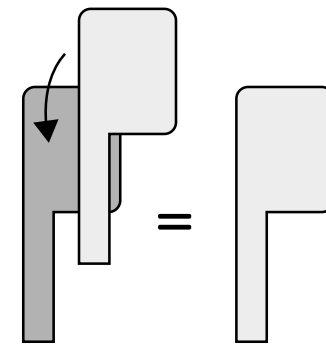
A few examples for inspiration:

Knitted stretch sensors <https://www.kobakant.at/DIY/?p=1762>

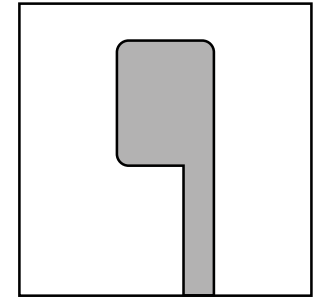
Circular knit inflation sensor: <https://www.kobakant.at/DIY/?p=1996>

Neoprene flex sensor: <https://www.kobakant.at/DIY/?p=20>

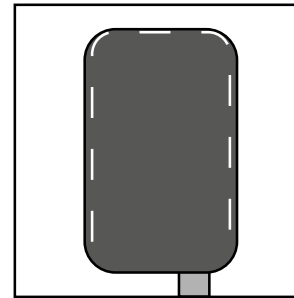
Embroidered potentiometer: <https://www.kobakant.at/DIY/?p=2331>



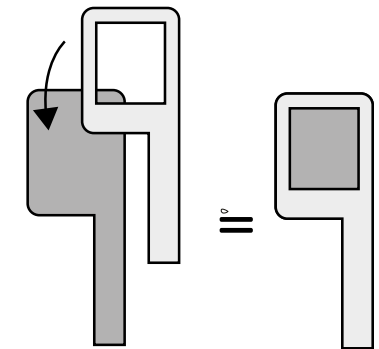
A. Fuse conductive fabric with iron-on fabric.



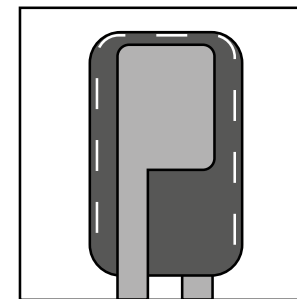
B. Flip and fuse onto non-conductive fabric.



C. Sew resistive velostat with non-conductive thread. Be careful to not stitch through conductive fabric.



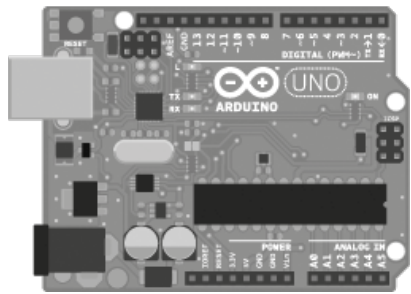
D. Fuse conductive fabric with iron-on fabric.



E. Turn iron on lowest setting. Flip and fuse onto resistive velostat.

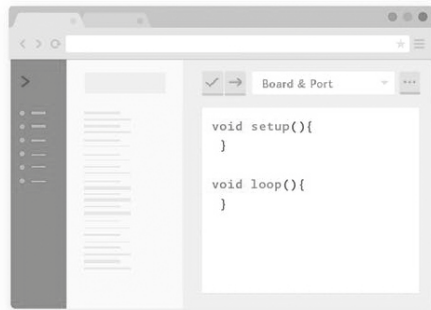
ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, playing a sound. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language and the Arduino Software (IDE).



HARDWARE

SOFTWARE &
PROGRAMMING
LANGUAGE



TO DROP CASUALLY INTO PARTY CONVERSATION:

Arduino was born at the Ivrea Interaction Design Institute in Italy in 2005 as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. The name Arduino comes from a bar in Ivrea, where some of the founders of the project used to meet.

HELLO WORLD

1. Install following the instructions for downloading and installing the Arduino IDE. <https://www.arduino.cc/en/Main/Software>
2. Open the IDE and connect your Arduino Uno board via USB.
3. In the Arduino IDE, check you have ticked 'Arduino/Genuino Uno' under menu point TOOLS > BOARD:
4. Under the same menu point 'TOOLS', go to 'Programmer:' and check you have ticked 'AVRISPmkII'.
5. Go to 'FILES > EXAMPLES > 01.BASICS', and open the built-in example sketch 'BLINK'.
6. Upload to your Arduino board by pressing the 'UPLOAD' button.
7. Wait a few seconds.
8. The onboard LED now blinks in intervals of 1 second.

★
CONGRATS!!!
YOU JUST SUCCESSFULLY RAN
YOUR FIRST PROGRAMME.
★

Built-in Examples are sketches included in the Arduino IDE, to open them click on the toolbar menu: File > Examples. These programs demonstrate all basic Arduino commands. An overview is here: <https://www.arduino.cc/en/Tutorial/BuiltInExamples>

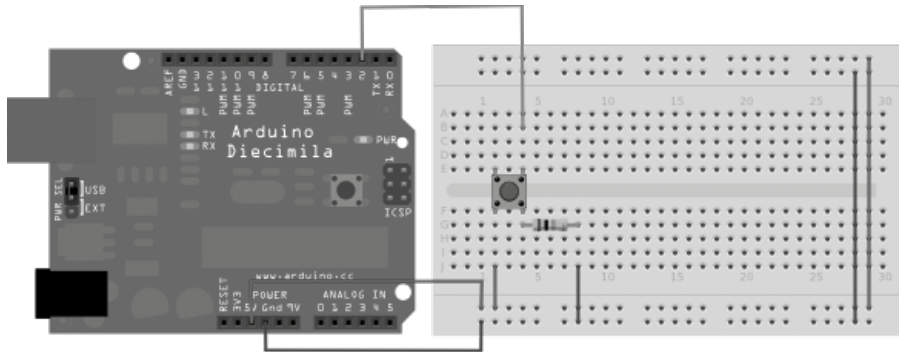
ARDUINO DigitalIn

from <https://www.arduino.cc/en/tutorial/button>

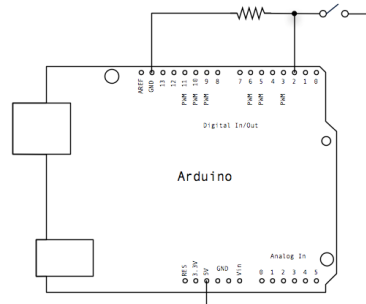
Pushbuttons or switches connect two points in a circuit when you press them (like the one you made from textiles previously). This example turns on the built-in LED on pin 13 when you press the button.

Hardware

- Arduino Board
- Momentary button or Switch
- 10K ohm resistor
- Jumper wires
- Breadboard



FOR REFERENCE - THIS IS THE CIRCUIT SCHEMATIC.



Open the example sketch '02.Digital > Button', and upload it. Now press the button to control your LED.

```
/*
  Button

  Turns on and off a light emitting diode(LED) connected to digital pin 13,
  when pressing a pushbutton attached to pin 2.

  The circuit:
  - LED attached from pin 13 to ground
  - pushbutton attached to pin 2 from +5V
  - 10K resistor attached to pin 2 from ground

  - Note: on most Arduinos there is already an LED on the board
    attached to pin 13.

  created 2005
  by DojoDave <http://www.0j0.org>
  modified 30 Aug 2011
  by Tom Igoe

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Button
*/

// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;     // the number of the LED pin

// variables will change:
int buttonState = 0;        // variable for reading the pushbutton status

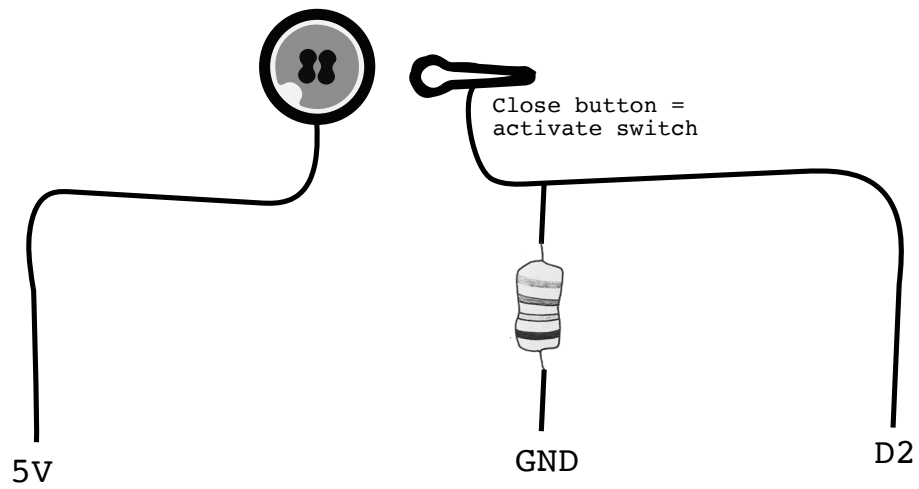
void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

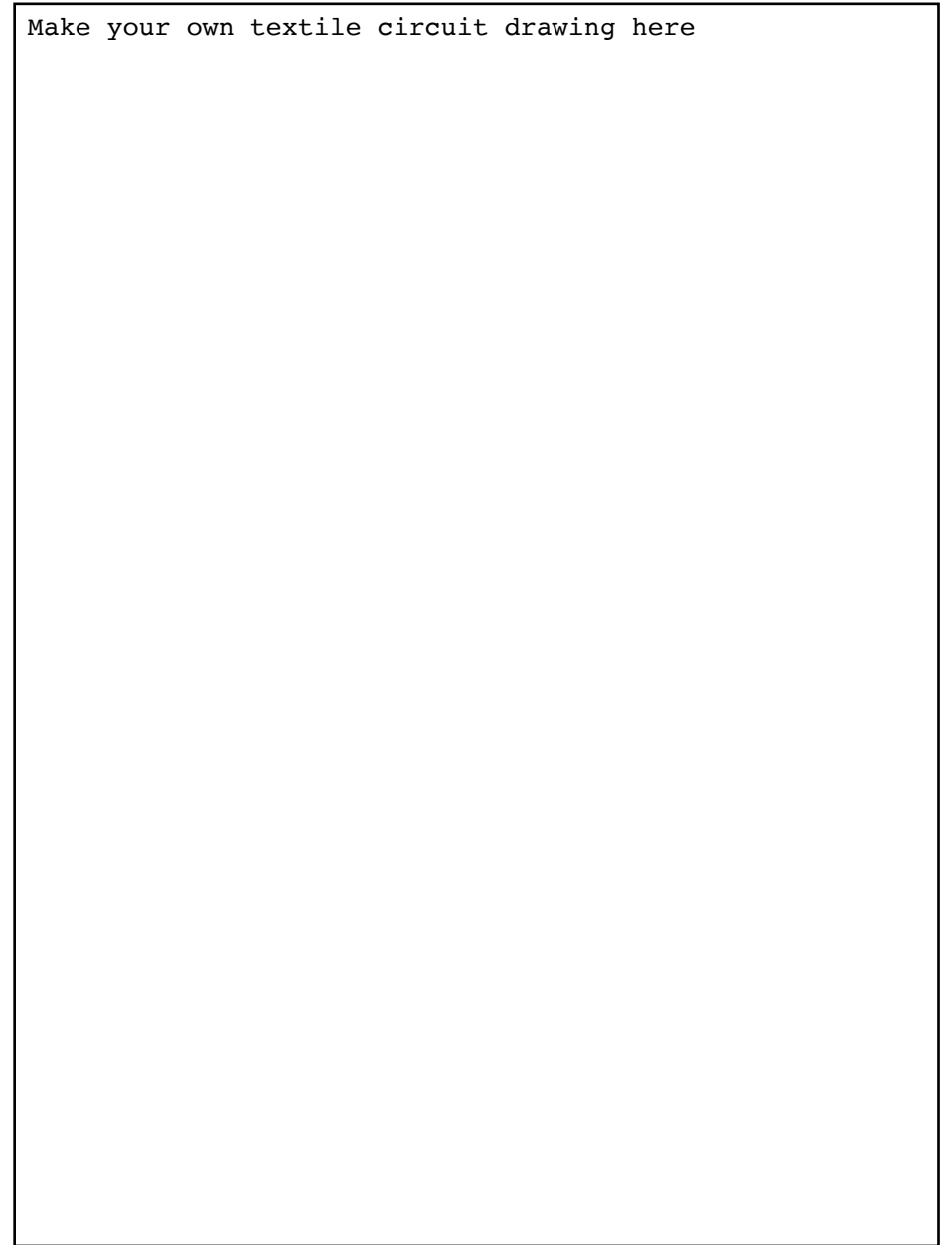
  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

ARDUINO DigitalIn

Use your textile switch instead.



Make your own textile circuit drawing here



ARDUINO

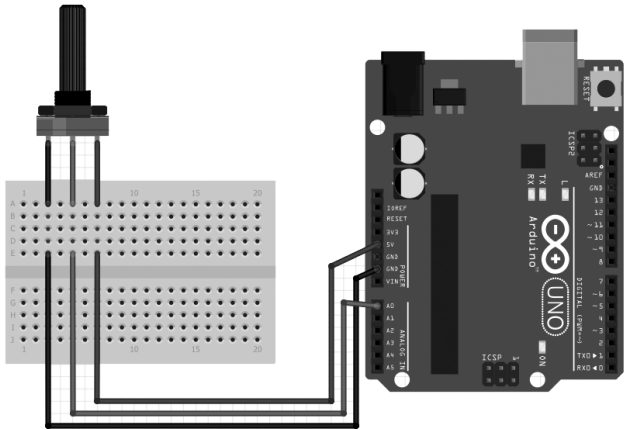
AnalogReadSerial

from <https://www.arduino.cc/en/Tutorial/AnalogReadSerial>

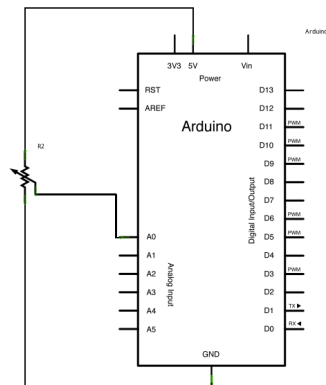
This example shows you how to read analog input from the physical world using a potentiometer.

Hardware

- Arduino Board
- 10k ohm Potentiometer
- Jumper wires
- breadboard



FOR REFERENCE - THIS IS THE CIRCUIT SCHEMATIC.



Open the example sketch '01.Basics > AnalogReadSerial', and upload it. Now use the potentiometer to change your sensor input, and watch the results on the serial monitor.

```

/*
  AnalogReadSerial

  Reads an analog input on pin 0, prints the result to the Serial Monitor.
  Graphical representation is available using Serial Plotter (Tools > Serial Plotter menu).
  Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/AnalogReadSerial
  */

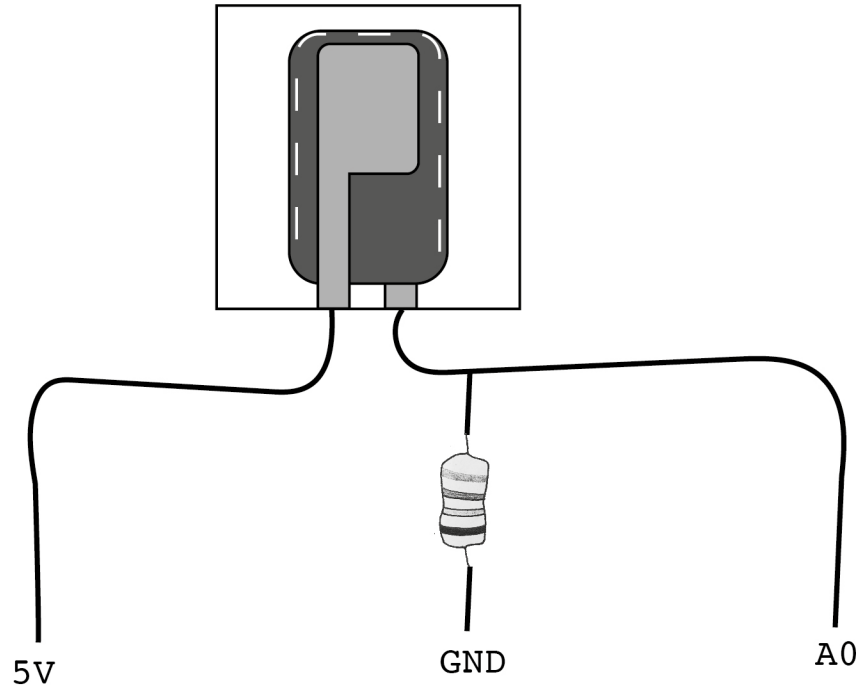
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}

```

ARDUINO AnalogRead

Use your analogue textile sensor instead.



Make your own textile circuit drawing here

REAL WORLD DATA TO COMPUTER SIGNALS

VOLTAGE DIVIDER

Your computer only understands digital signals (with a Bit being the smallest unit available - 0 or 1). Therefore, we need a converter to translate the "realworld" data coming from your textile sensors into digital data. This is called an "**Analog-to-Digital Converter**" - **ADC** in short. This is done by your Arduino board.

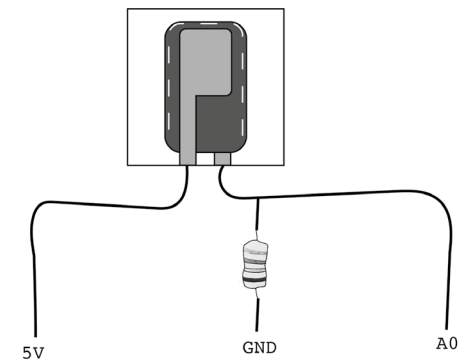
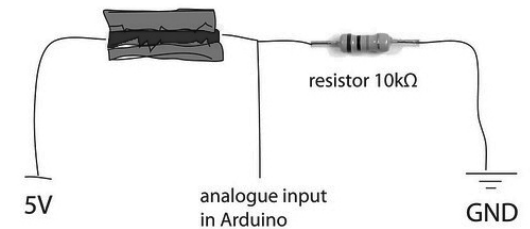
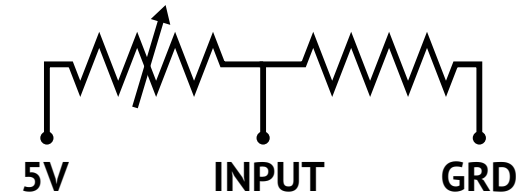
But: The output of your sensor is electrical resistance, which is an inherent material property. Arduino does not understand it. Resistance needs to be converted to voltage first.

THAT IS GOOD TO KNOW... BUT how do we do this??

BY USING A VOLTAGE DIVIDER!

You can divide the voltage by using 2 resistors. As the ration between two resistors changes, the voltage you get in between them changes. This means we can send Arduino a change in voltage, which it can understand.

THESE ALL MEAN THE SAME!



GRAPHICAL PROGRAMMING

```
module Main where

main :: IO ()
main = putStrLn "Hello, World!"
```

```
print "Hello, World!\n";
```

```
classdef hello
  methods
    function greet(this)
      disp('Hello, World')
    end
  end
end
```

```
#include <iostream>

int main()
{
  std::cout << "Hello,
world!\n";
  return 0;
}
```

```
00110100 00101010 11001111 00101010
00101111 11101010 01101010 11010101
00101011 01101010 00101010 11111010
10101001 00101111 01010111 01010111
```

```
program HelloWorld(output);
begin
  Write('Hello, world!')
end.
```

```
b8 21 0a 00 00 #moving "\n" into eax
a3 0c 10 00 06 #moving eax into first memory location
b8 6f 72 6c 64 #moving "orld" into eax
a3 08 10 00 06 #moving eax into next memory location
b8 6f 2c 20 57 #moving "o, W" into eax
a3 04 10 00 06 #moving eax into next memory location
b8 48 65 6c 6c #moving "Hell" into eax
a3 00 10 00 06 #moving eax into next memory location
b9 00 10 00 06 #moving pointer to start of memory
ba 10 00 00 00 #moving string size into edx
bb 01 00 00 00 #moving "stdout" number to ebx
b8 04 00 00 00 #moving "print out" syscall number
cd 80 #calling the linux kernel to execute
b8 01 00 00 00 #moving "sys_exit" call number
cd 80 #executing it via linux syscall
```

Traditionally, computer programmers used text-based programming languages to write applications (see page left). The programmer would write lines of code into a file, and then run it afterwards to see the results.

Many sound or visual artists, as well as other non-programmers, find this a difficult and non-intuitive method of creating things however.

I LOVE THIS VARIETY OF LANGUAGES. GREAT YOU CAN SAY THE SAME THING IN SO MANY DIFFERENT WAYS.

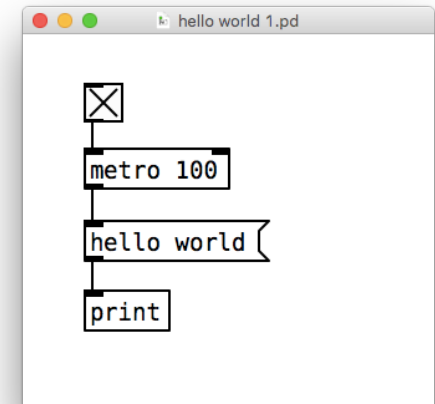
BUT I DO NOT UNDERSTAND A THING! CAN SOMEONE MAKE A DRAWING PLEASE??

Real Time Graphical Programming

Pure Data, on the other hand, is a graphical programming environment. What this means is that the lines of code, which describe the functions of a program and how they interact, have been replaced with visual objects which can be manipulated on-screen. Users of Pure Data can create new programs (patches) by placing functions (objects, see also next page) on the screen. They can change the way these objects behave by sending them messages and by connecting them together in different ways by drawing lines between them.

Pure Data (or just: Pd) is an open source visual programming environment that runs on anything from personal computers to embedded devices (i.e. Raspberry Pi). It was originally developed by Miller Puckette at IRCAM.

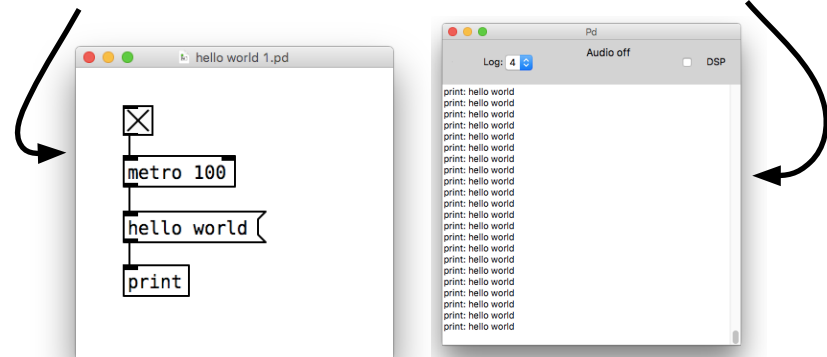
Pd enables musicians, visual artists, performers, researchers, and developers to create software graphically without writing lines of code. Pd can be used to process and generate sound, video, 2D/3D graphics, and interface sensors, input devices, and MIDI.



PURE DATA Intro

PATCH Pure Data files are called patches. A patch is identical to a programme.

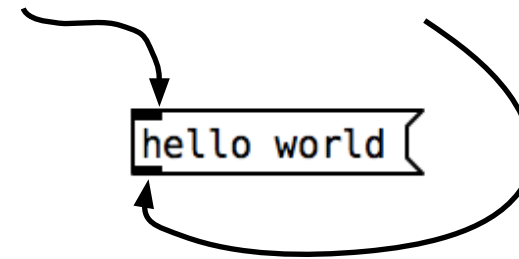
This programme prints "hello world" in the Pd window.



COMMENTS A "comment" is simply a way of making a note to yourself so that you (or someone else) can understand what you were trying to do later on. You can make as few or as many as you want, and they have no effect on the patch itself.

this is a comment

INLETS AND OUTLETS



Data flow is vertical, from the top of the patch to the bottom. Data goes into inlets, and out of outlets.

OBJECTS Objects are the "vocabulary" of Pd. Use the "Put" menu to place an "Object" in your patch. Click on the patch to drop the object in its place. You will see a box made of a broken blue line, with a flashing cursor inside indicating that you should type something there.

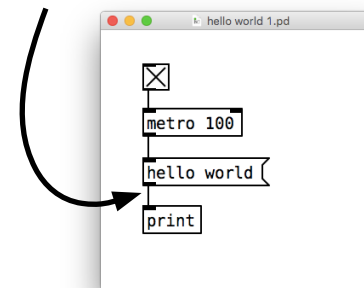
The more names of objects you know, the more complicated things you can do with Pd. If you type the word "print" inside this object and click again outside the box, you will create the [print] object.

Return to the "Put" menu, and this time place a "Number" in your patch. Notice that the shape of the number box is different from the shape of the object box.

print

4

CABLES



To make a connection between two boxes (objects, messages, etc), click on any outlet of the first one, and drag the cable toward an inlet of the second one, and release.

MESSAGES The "Message" box is used to store and send information to other objects, and can contain numbers or text. It also has a unique shape, which resembles an envelope like you would use to send a letter.

hello world

LOCK AND UNLOCK THE PATCH

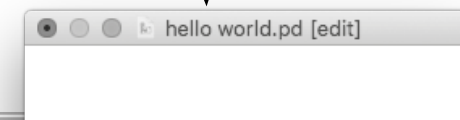
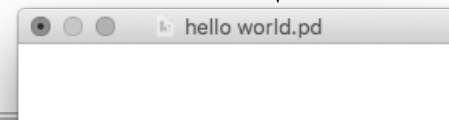
To **edit** a patch, it needs to be unlocked. To **run** a patch, it needs to be locked.

To lock and unlock, use the key short cut **command + e**.

Status report in window bar

locked

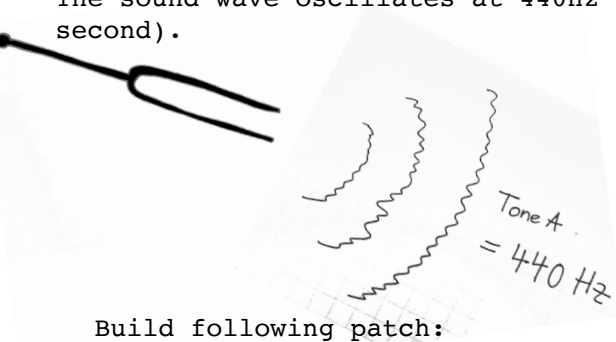
unlocked



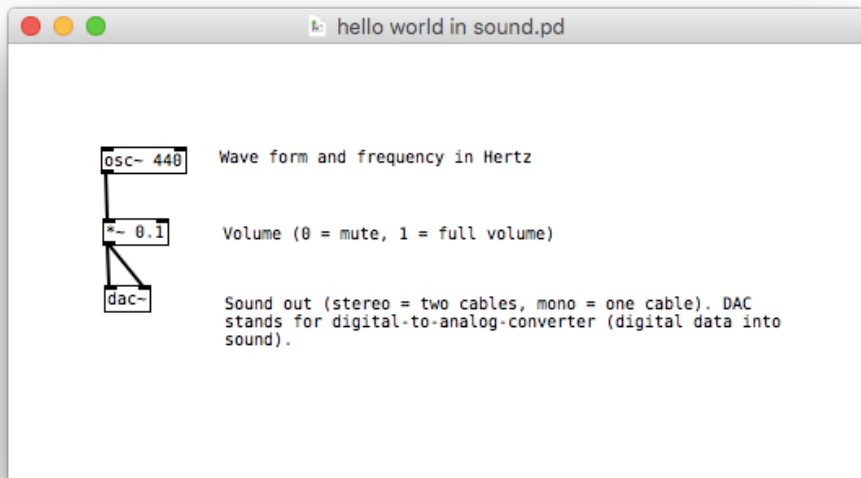
PURE DATA

HELLO WORLD (440HZ)

Since we want to explore sound, we will introduce you to the "hello world" equivalent in the sound world. This is usually the tone A - the sound of a tuning fork! The sound wave oscillates at 440Hz (sound wave cycles per second).



Build following patch:



Remember to turn the DSP on so that you can hear the audio. DSP stands for 'digital signal processing', and in our case, turns on the sound.

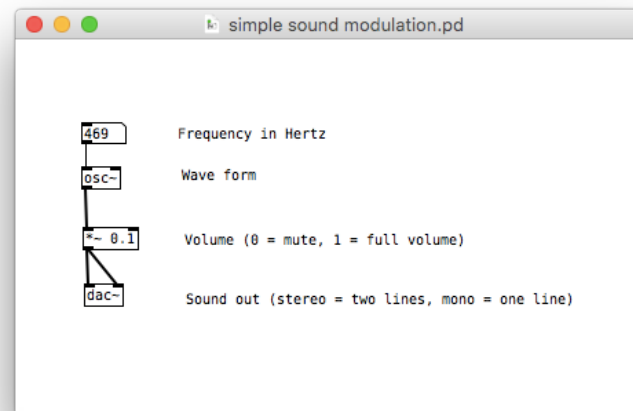
Audio vs Message Cables

In the previous example, notice the difference between the cable for messages, which is thin, and the cable for audio in this example, which is thicker. Messages can be sent to audio objects (those with a ~ in their name), but usually audio cannot be sent to message objects (those without a ~ in their name). Attempting to do so will cause Pd to print "error: can't connect signal outlet to control inlet", and it will not allow the connection to be made.

Oscillator

The Sine Wave Oscillator (osc~) makes a pure tone with no harmonics. Try 'phasor~' instead and listen to the difference.

Now change your patch to:



Simple sound modulation (pitch)

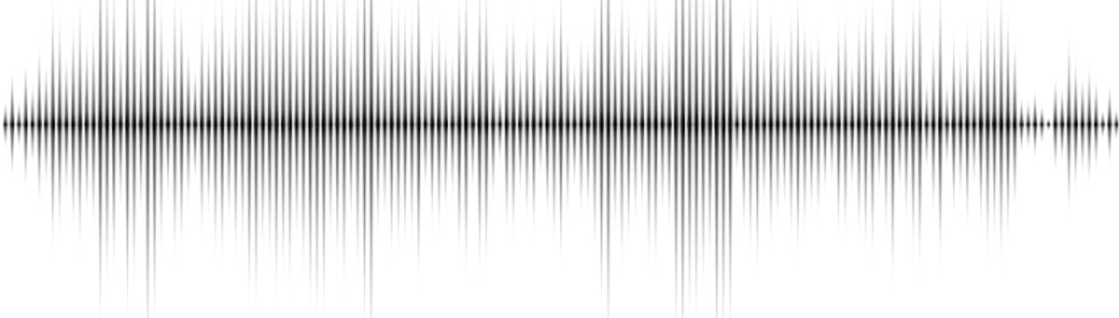
You can use a number object to define and vary the input to the osc~ object: Lock your patch. Now use the mouse to run the number in the number box up and down - and the pitch of your oscillator varies.

CONGRATS! YOUR FIRST SYNTHESIZER. REWARD: LISTEN TO THE DR WHO THEME TUNE, CREATED BY ELECTRONIC MUSIC PIONEER DELIA DERBYSHIRE IN 1963!



PURE DATA

Play sound file



A sound sample is a small section (usually a couple seconds long) of recorded sound.

In music, sampling is the reuse of a portion (or sample) of a sound recording in another recording. Samples may comprise rhythm, melody, speech, or other sounds. Sampling was fundamental in the foundation of hip hop music, but has had and still has wide reaching influence on other music styles and experimental sound art.



Drum machine Akai MPC60, 1988

Live coder Alex McLean, using samples in TidleCycle.

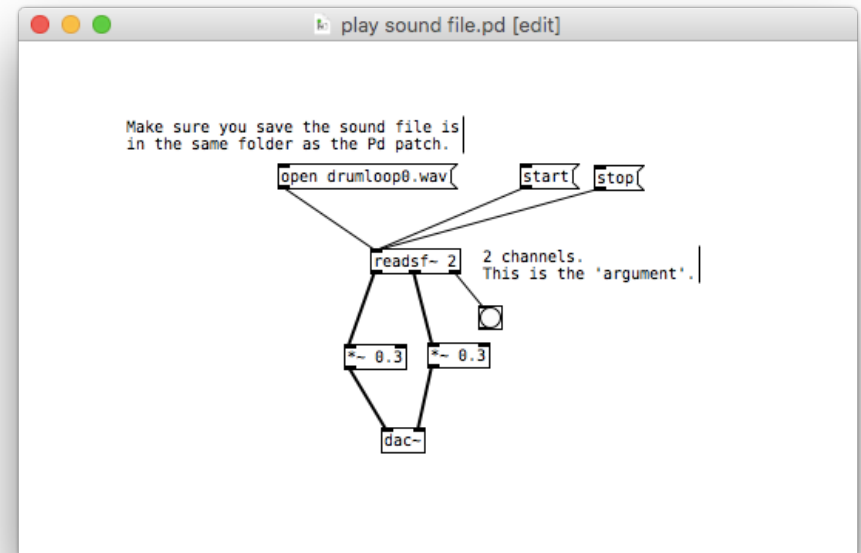


40 Photo: Under your skin.

In Pd, you can select, play, save, and edit sound files in many ways.

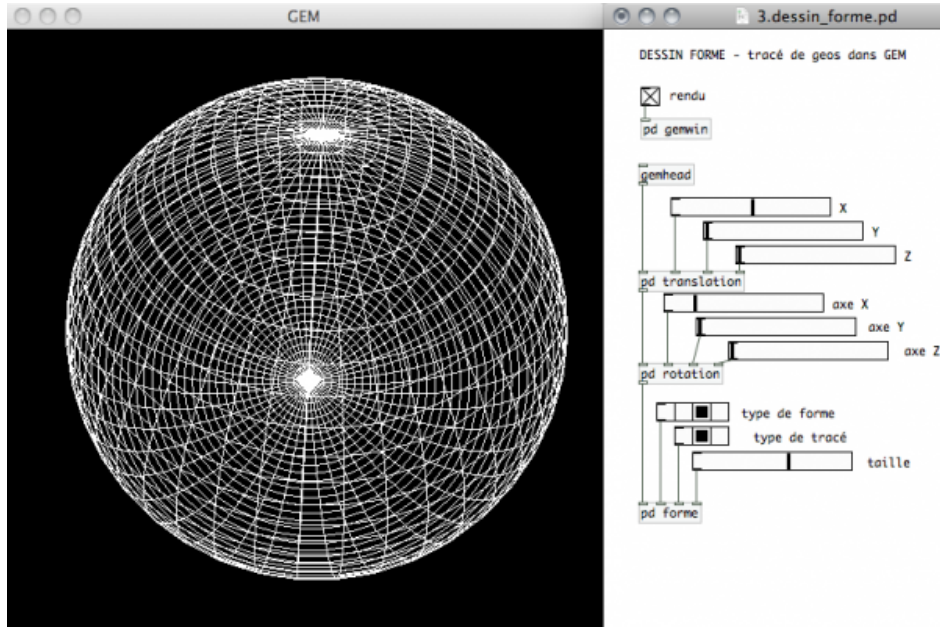
Play a sound file from your computer

Sound files that are on an external storage device like a hard disk can be read - that is, played back - in Pd with the object 'readsf-'. You use the messages 'start' and 'stop' (you could also use '1' and '0'). Enter the number of channels as the argument. The rightmost output sends a bang when the end of a file is reached.



PURE DATA

GEM Graphics Environment for Multimedia



GEM is a visual programming environment, which allows you to easier control visuals in Pd, e.g. videos, photos, animations.

GEM is a collection of **externals** which allow the user to create OpenGL graphics within Pd.

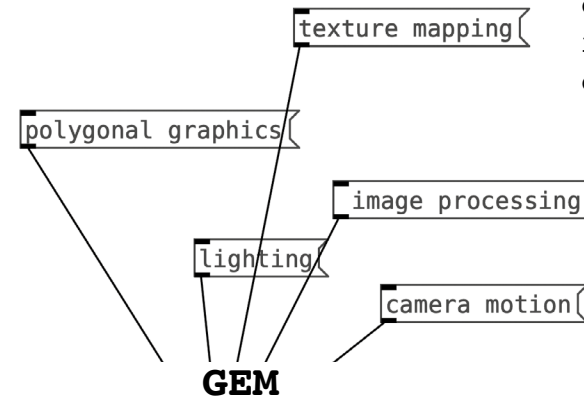
You can also combine audio and graphics, controlling one medium from another, or externally.

Important!

GEM is NOT an application, it's a library.

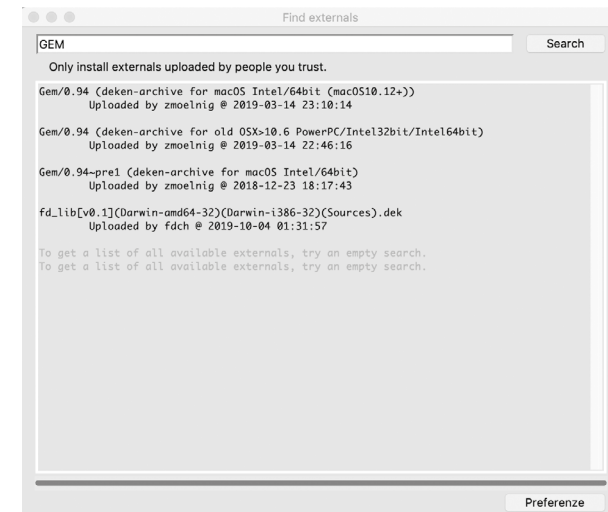
What's a library?

In computer science, a library is a collection of resources used by computer programs.



How do I install GEM?

Start Pure Data > Help > Find externals > Search "GEM". Click the newest version correct for your operating system, and click 'yes' to install.



PURE DATA

Serial read

If we want to read values from Arduino in Pure Data, we need to connect them. We can connect them through a USB cable and read the values through the USB serial port.



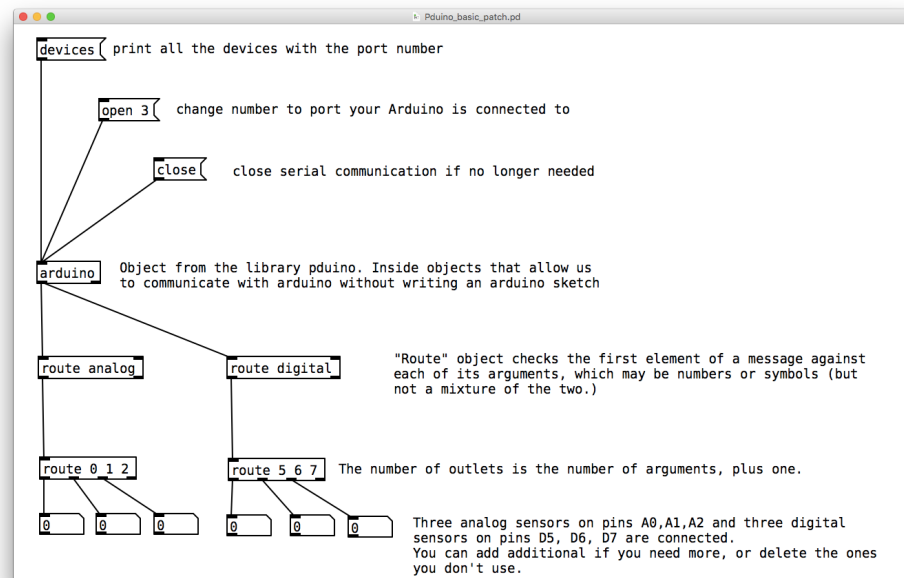
To read and write data through the serial port we need a library called 'Pduino'. Objects in Pduino allow you to use Arduino boards via the Firmata protocol.

How to install Pduino:

1. Start Pure Data > Help > Find externals > Search "**comport**". Click the newest version correct for your operating system, and click 'yes' to install.
2. Start Pure Data > Help > Find externals > Search "**Pduino**". Click the newest version correct for your operating system, and click 'yes' to install.
3. Connect an Arduino board to your computer. Start Arduino software > Examples > Firmata > AllInputsFirmata. Upload to the board. Once successfully uploaded, quit the Arduino software.

How to use Pduino

1. Build following patch:



2. Print the list of the devices, and find the correct port number in the Pd window (Mac: /dev/cu.usbmodem1411, Windows: COM3).
3. Change the number in the 'open' message to select the right port.
4. Lock the patch and click 'open' to open the port.
5. If everything was done correctly and your Arduino is connected, you will see now the sensor readings.

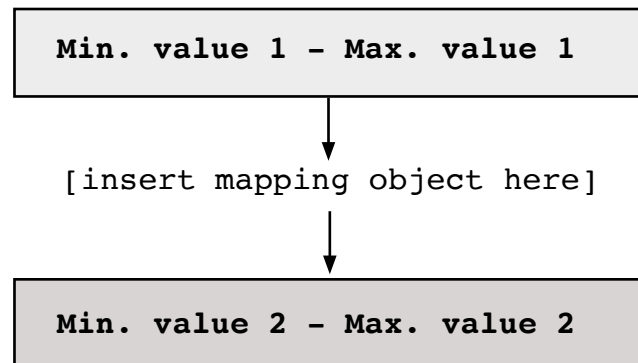
PURE DATA

Mapping sensor data

In computing, data mapping is the process of translating one set of numbers into another. This is useful for example, when the input range of your sensor is not quite as extensive as needed - with mapping you can extend it. Or if the data coming from your sensor jittery - with mapping you can smooth it.

But above all, mapping allows you to be really creative with what you want the output to be as a response to the sensor input.

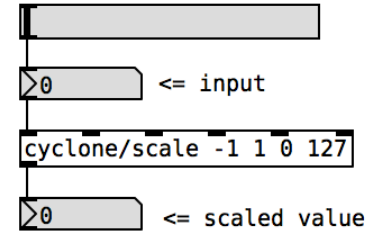
This is also, where digital technology provides you with the freedom to design. For example, in your analogue switch circuit (page 19), the LED switches on, once you press the button, and there is no possibility to - let's say - do it in reverse. Using an Arduino however, allows you to interpret the pressing of the button in any possible way. You could have the LED on, and the button switches it off. Or you have a pressure sensor input of 0 - 1023, mapping it to five different sounds. The pressure sensor of your neighbour might have an input range though of 327 - 965, but they can still map the pressure to the same five sounds.



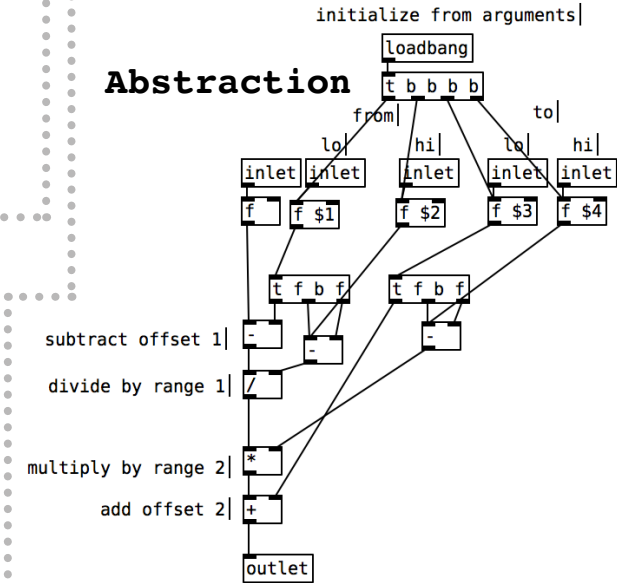
Example mapping tools in Pure Data:

From Cyclone Library

(install via Find Externals)



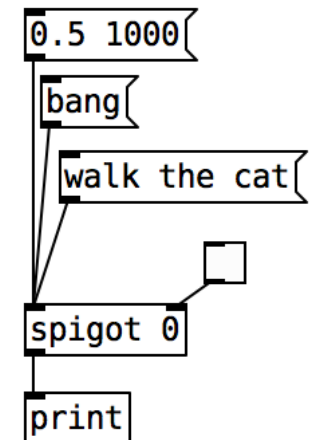
Abstraction



Spigot object

Pass or block messages

Spigot passes messages from its left inlet to its outlet, as long as a nonzero number is sent to its right inlet. When its right inlet gets zero, incoming messages are "blocked," i.e., ignored.



RESOURCES

Materials and components

Local Electronic shops

- Segor <http://www.segor.de/#/>, Berlin-Charlottenburg
- Conrad <https://www.conrad.de>, Berlin-Kreuzberg

Local Haberdashery & Art/Crafts Supply

- Hobbyshop Rüter <https://www.hobbyshop.de>, Berlin-Schöneberg
- Modulor <https://www.modulor.de> Berlin-Kreuzberg

Online Electronics

- EXP-Tech <https://www.exp-tech.de>
- Watterott Electronic <https://www.watterott.com>
- Digi Key <https://www.digikey.de>
- Conrad <https://www.conrad.de>
- Paradisetronic <https://paradisetrionic.com>
- Eckstein Komponente <https://eckstein-shop.de>
- Pimoroni <https://shop.pimoroni.de>

Conductive fabrics (online)

- Statex <https://www.statex.de/en/fabrics/>
- LesseMF <http://www.lessemf.com/fabric.html> (U.S.)
- Eeonyx <http://eeonyx.com> (U.S.): Conductive Threads and Yarns
- Statex <https://www.statex.de/en/fibres-and-yarns/>
- Karl Grimm <http://www.karl-grimm.com>,
<http://www.flickr.com/photos/plusea/3450531172/sizes/l/in/photostream/>
<http://www.flickr.com/photos/plusea/3449716377/sizes/l/in/photostream/>
<http://www.flickr.com/photos/plusea/3450531656/sizes/l/in/photostream/>
<http://www.flickr.com/photos/plusea/3449716527/sizes/l/in/photostream/>

- Bakaert <https://www.bekaert.com/en/> (BE)
- Bart Francis, <https://bart-francis.be/index.php?action=home&lang=EN> (BE)
- Make your own <https://www.instructables.com/id/Conductive-Thread-Wind-up/>

Pressure-sensitive materials

- Velostat <http://multimedia.3m.com/mws/media/3052400/stock-de-hojas-velostattermoconformables.PDF>
- Anti-static conductive foam, e.g. <https://www.conrad.de/de/esd-schaumstoff-1-x-b-x-h-300-x-300-x-6-mm-ableitfaehig-esd-kennbuchstabe-d-wolfgang-warmbier-172367.html>

Conductive inks:

- Conductive paint, BareConductive <https://www.bareconductive.com/shop/electric-paint-50ml/> (UK)
- CuPro-Cote® PAINT, LesseMF, <http://www.lessemf.com/292.html> (U.S.)

RESOURCES

Learning & tutorials on the web

E-textiles & Crafts:

Most comprehensive, including many material, code and hardware resources:

- KOBAKANT, How to Get What You Want: <https://www.kobakant.at/DIY/>

How-to-etextiles videos:

- Instructables: <https://www.instructables.com/id/How-To-videos-for-eTextiles-soft-circuits-and-we/>

Fabricademy:

- eTextiles + Wearables I, Slides Liza Stark (The Soft Circuiteer): <http://thesoftcircuiteer.net/fabricademy-etextiles-wearables-i/>

Electronics & Software:

Intro electronics:

- Sparkfun: Tutorials covering a range of stuff, including the basics of electricity and electronics <https://learn.sparkfun.com/tutorials/where-do-i-start/all>
- Makerspaces: Overview components and tools <https://www.makerspaces.com/basic-electronics/>

Arduino:

- Arduino tutorial page linked to different directions of learning (e.g. example projects, Arduino foundations etc) <https://www.arduino.cc/en/Tutorial/HomePage>

Pure Data: <https://puredata.info>

- Good tutorial (book) in English: <http://write.flossmanuals.net/pure-data/introduction2/>
- Good tutorial (book) in German: <http://pd-tutorial.com/german/index.html>

RESOURCES

Literature Theory/Context

Cork, C. R. (2015). Conductive fibres for electronic textiles. In *Electronic Textiles: Smart Fabrics and Wearable Technology* (1st Editio, pp. 3–20).

Sennett, R. (2008). *The Craftsman*. London: Allen Lane.

Schwartzman, M. (2011). *See Yourself Sensing: Redefining Human Perception*. London: Black Dog Publishing Limited.

Ryan, S. E. (2014). *Garments of Paradise: wearable discourses in the digital age*. Cambridge: The MIT Press.

Pye, D. (1995). *The Nature and Art of Workmanship*. New York: Bloomsbury Publishing Inc.

THANK YOU to the e-textiles and open source communities who generously and capably share their knowledge and resources.

Especially:

- for the zine and teaching inspirations provided by the e-textiles community, including Becky Stewart, Hannah Perner-Wilson, Mika Satomi, Liza Stark, Becca Rose, and all participants in e-textile summercamp 2019.
- to the Pd developer community. We used parts of original text from the Pure Data FLOSS manual.
- to the Arduino community. We used original text and illustrations from arduino.cc.
- to the Sparkfun team. We used parts of original text for the electronics introduction.



CC BY-SA 4.0: This license allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator. The license allows for commercial use. If you remix, adapt, or build upon the material, you must license the modified material under identical terms.

Version 2.0 for WS20/21 seminar: Crafting Textile Data
Berit Greinke and Giorgia Petri

Photos and illustrations (if not stated otherwise):
Berit Greinke

Contact: b.greinke@udk-berlin.de



Universität der Künste Berlin

**WEARABLE
COMPUTING**

