FEMINIST HARDWARE: MAKING PRINTED CIRCUIT BOARDS WITH NATURAL CLAY

BY PATRÍCIA J. REIS AND STEFANIE WUSCHITZ

INTRODUCTION

It is an open secret that the hardware in our smart devices contains not only plastics but also conflict minerals such as tungsten, tin, tantalum, silver and gold. We are investigating alternative hardware from locally sourced materials – so-called ethical hardware – to speculate and develop renewable practices for the benefit of both nature and humans.

We are exploring different materials: sentient, low-impact, non-toxic, fairly traded, recycled and urban-mined means of production. We aim to challenge the common PCB (printed circuit board) economies in an artistic, creative, positive and responsible way, applying feminist hacking as an artistic methodology and critical framework.

The aim of this tutorial is to provide a step-by-step manual to share the experience acquired through practice-based research in the process of making printed circuit boards (PCBs). This tutorial focuses on building PCBs with natural clay using recycled silver as the main electrical conductor. Our PCB is an artistic project but also an interactive tool for creatives, hackers and artists. It works as a microcontroller board that can control digital and analogue sensors as inputs and speakers, LEDs and motors as outputs.

We joined forces with Daniel Schatzmayr to develop the electric circuit and the code programmed to the chip. Our intention, however, is to encourage you to play with the hardware and code yourself, with the option to modify it if you feel like it. You can do this by downloading our code from GitHub, editing and uploading it via the Arduino IDE. Our project is all open sourced, and we are happy to receive feedback and hear about your own experiences with it.

This tutorial has **five parts**, each containing several steps:
PART 1: Circuit and the 3D printed 'stamp'
PART 2: Clay (collecting soil, modelling the boards, painting the circuit and firing it in an open fire)
PART 3: Programming (how to use the bootloader to upload a program to the ATmega chip)
PART 4: Ethical electronics (soldering the ethical hardware electronic components)
PART 5: Powering and testing

PART 1: Circuit and the 3D printed 'stamp'

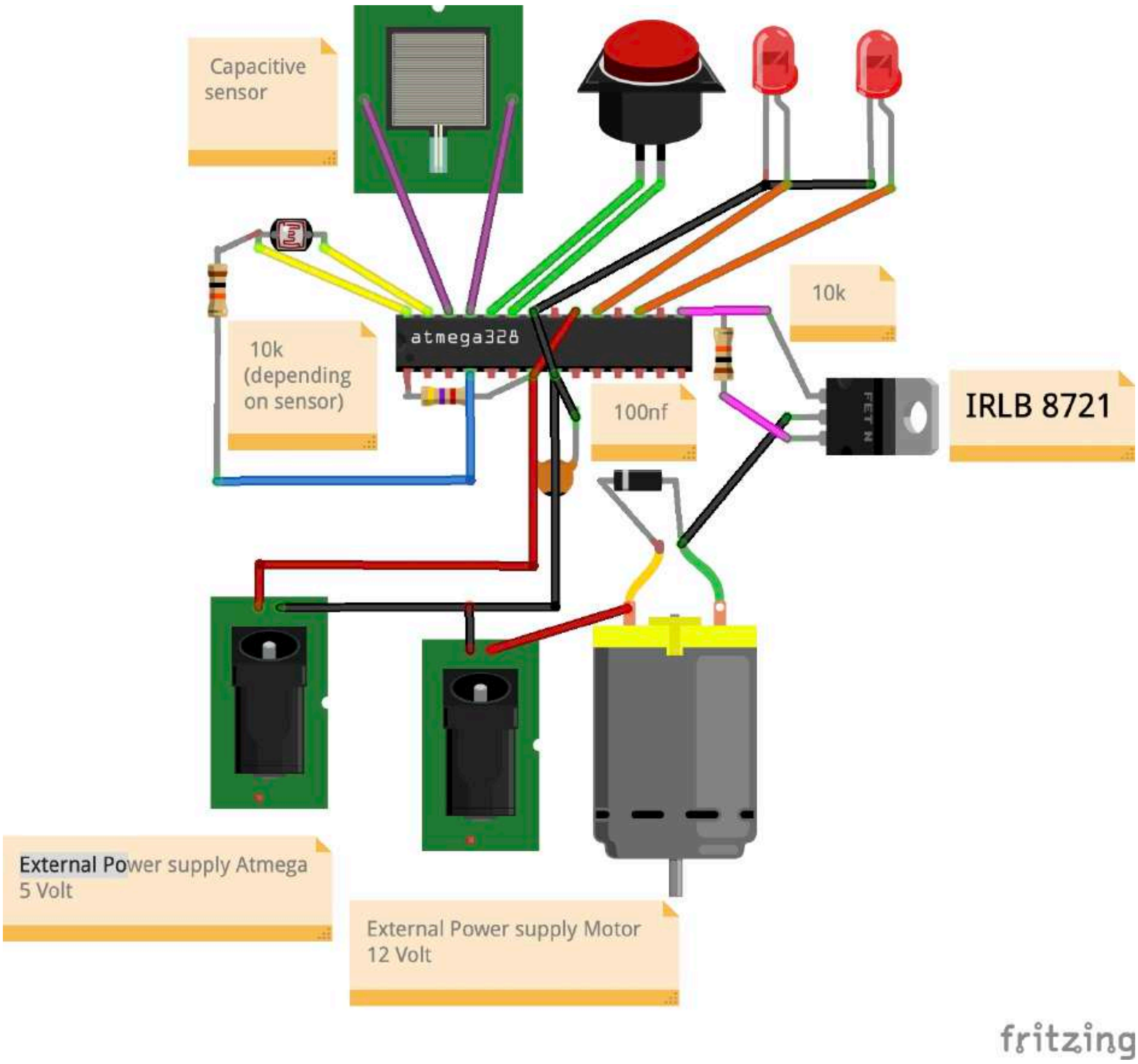
Our initial idea was to develop a microcontroller PCB that could work with the **ATmega328P** chip, which is commonly used in the famous Arduino Uno board (or **Arduina** board, as some feminists call it). Why this chip? Because we are part of a community hackerspace – Mz* Baltazar's Lab (a feminist hacklab and artist-run independent space based in Vienna, Austria) – and the Arduino Uno has been our favourite microcontroller for the past 12 years. After using it in many prototypes, artworks and workshops, we had several malfunctioning Arduino boards left over. But their chips were still working, so the idea was to reuse the chips in our new project.
The second challenge was to come up with an electric circuit that would allow us to receive several forms of **input signal** (analogue and digital sensors) and generate a variety of **output signals** (to control LEDs, motors and speakers). Using the smallest amount of ATmega chip pins possible, to simplify the circuit, we drew a diagram using Fritzing for the following functions:

Input:

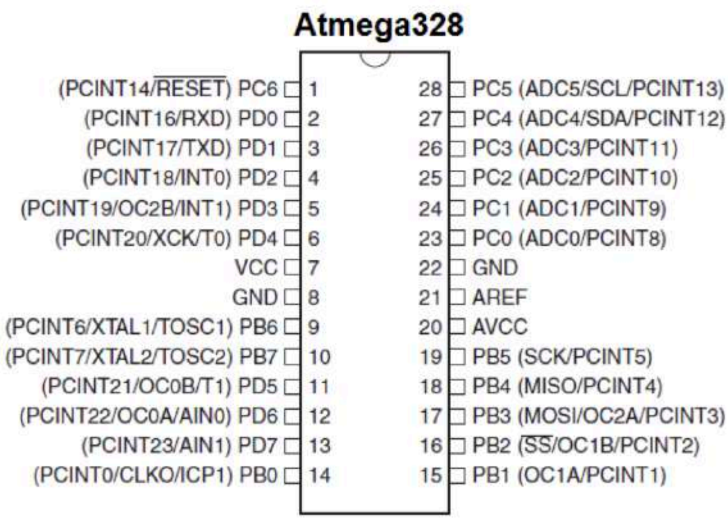
- Capacitive sensor – ATmega pins A3 + A2
- Analogue sensor with a voltage divider – ATmega pins A4 + A5
- Digital button – ATmega pins A1 + A2 + 2

Output:

- 2 x LEDs or piezo buzzers – ATmega pins 13 + 11
- DC motor with a transistor circuit – ATmega pin 9

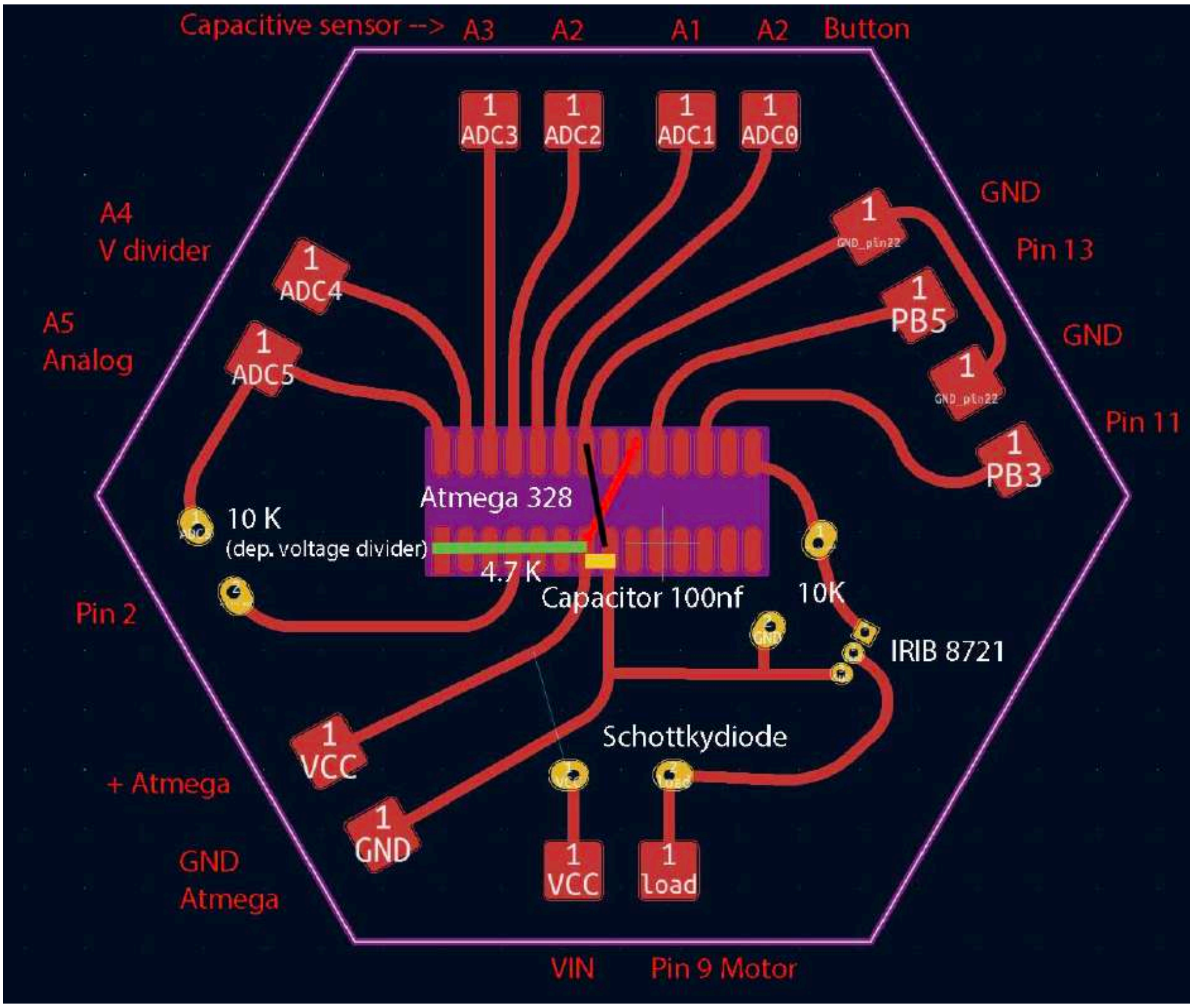


The illustration below helps to explain the **ATmega328** pin-out. We will provide more details about the electronic components in **PART 4: Ethical electronics (soldering the electronic components)**.



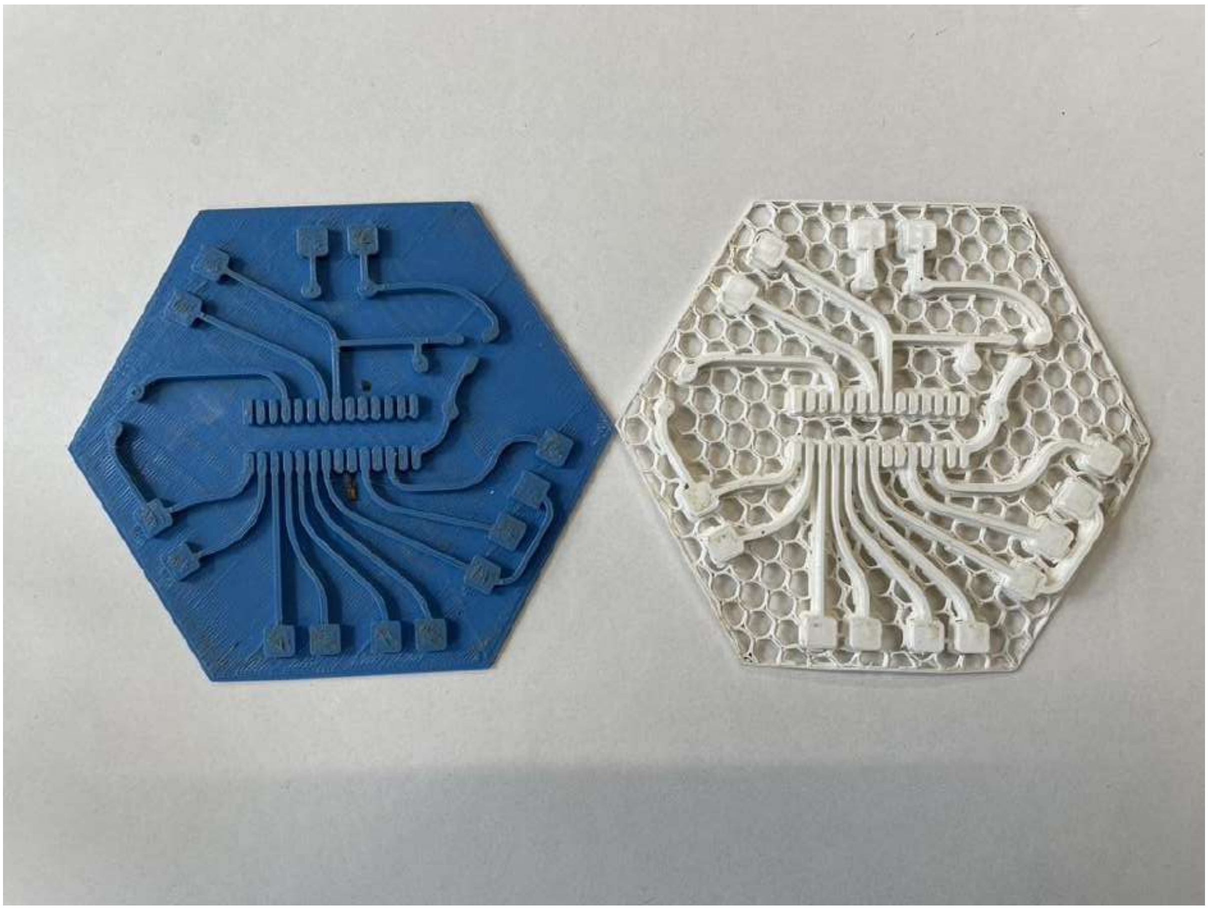
The next step was to **simplify the circuit** in order to reduce it to a **single printable layer**. To avoid having multiple layers, we decided to solder some of the connections and electronic parts directly to the top of the **ATmega328** (which is also explained in more detail later). These connect **VCC pin 7** with **AVCC pin 20**, as well as **ground pin 8** and **ground pin 22**. We also added a bridge from the **RESET pin to VCC** through a 4.7 K resistor.

Below is a diagram of the circuit in a hexagon shape.



With this insight, we were able to design a **3D printed 'stamp'** using a recycled polypropylene filament. This process took a while, because we had to consider the clay's changed size after drying and firing. All clay shrinks considerably during the drying and firing process. We estimate that there is a **5% shrinking** rate, but this might vary depending on the clay you are working with. We also experimented with the depth of the circuit tracks and realised that the imprint should ideally be **1.2 mm deep**.

Below are some pictures of some of the 'stamps' and the files so that you can **3D print** them yourself.



PART 2: Clay (collecting soil, modelling the boards, painting the circuit and firing it in an open fire)

When we began the research on alternative materials for circuit building, we separated them into two categories: **conducting and insulating materials**.

To build the **base of our PCB,** we needed insulating, sustainable and robust materials (eggshells? wood plates? wax? ceramics?). We immediately chose ceramics, specifically **porcelain**, as it already plays an important role in electronic components such as capacitors, piezo, resistors, and so on. Porcelain is an industrially made material comprising **kaolin** (the main ingredient that makes it plastic and white) and **stone pottery** (the second ingredient that makes porcelain translucent and hard). Both are well-known commodities that are prospected and mined around the world, on a small scale in Europe and a larger scale in China, Brazil, South Africa and Vietnam (among others). In pottery, porcelain, also known as **china clay**, is a very delicate and sensitive material (we could say it has its own agency), more difficult to control than other industrial clays. Also, along with the other harder and more resistant stoneware clays, it usually requires higher firing temperatures, in two stages: a first firing known as a ceramic **bisque of *c.* 1000 ºC;** and a **glazing firing around 1200 ºC** in an electric kiln. During our initial experiments with porcelain, we were immediately aware that the higher temperatures, and therefore electric consumption, were not compatible with our standards for **ethical hardware**.

It was when we were struggling with the question of how to manufacture clay in low-energy and low-impact ways that we came across the work of **Heinz Lackinger**, a pottery crafter in Donnerskirchen, Burgenland, Austria, who works with prehistoric techniques of firing clay in an open wood fire. Instead of using sophisticated machines, he uses a simple hole in the grounds of his 18th-century house. We had the privilege of spending two days with this skilled craftsman, learning how to identify and collect the clay, and how to model and fire it using old, dry branches collected from the forest ground. If the clay is collected with awareness of its many qualities, and in small quantities only, this process can be defined as **100% fair trade** and congruent with locally sourced modes of hardware production. We owe the knowledge required for the following steps to Heinz Lackinger's generous knowledge transfer during his workshop, and to our own experiments later applying this technique in the making of natural clay PCB boards.

A – Collecting the clay

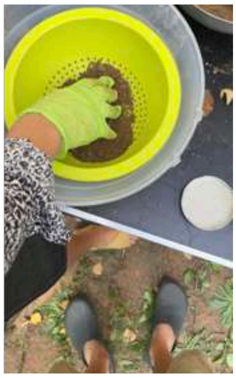
As we walk through the forest on an early autumn morning, the air is slightly humid. We are a small group of participants and Heinz is leading the way, carrying only a bottle of water in his hand. He grabs a small amount of soil and asks: **'Is this soil clay?'** From his bottle, he spills some water on the soil and starts to gently mix it with his fingers. The soil colour mutates from dark to reddish brown, and when it is massaged between the fingers one senses a certain increase in plasticity. It is definitely clay! We have **no lab tools** in the forest, only our human **sensory system**. Heinz explains that all landscapes are different: identifying clay in Burgenland might be different to identifying it in, for example, Brazil. Nevertheless, soil with argil properties is finer than sand and different to mud or soil – it feels soft and malleable in the hand when water is added. Most of our boards were made with different clay samples from this area, with varying degrees of argil properties. We observed that the clay with fewer argil properties tends to shrink more and is less resistant and waterproof.



B – Cleaning the clay

We collected our clay at the beginning of autumn in dry weather. The soil is mainly dry at this time, but it doesn't consist of argil alone: you will find small stones, plants, even small insects. When the clay is that dry, the easiest way to clean it is using a net that retains the undesirable waste. Our favourite tool for this is a standard **kitchen colander**. The waste collected should be given back to the earth, put back in the ground.





You will end up with a fine powder that needs some water. We calculated an average of **100 ml of water per 1 kg** of fine powder. Mixing it is just like blending flour and water, but without the inconvenient grumps. You should end up with something like a ball of clay after 10 minutes of massaging. It is important that the clay sticks together and all the air is removed.



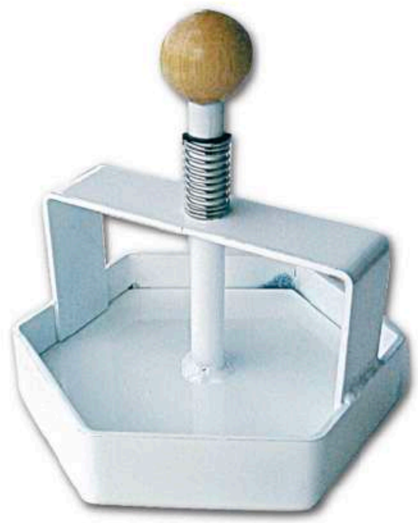


C – Modelling the clay

For the shape of our PCB board, we used a **hexagon tile cutter measuring 10 x 10 cm,** which can be bought in most ceramic shops. We chose a hexagon shape, as the tile form is not mandatory; but you can pick **any shape** and thickness you desire, as long as it maintains a printable surface of ***c.* 10 x 10 cm**. We chose this shape and format in the hope of assembling the boards as tiles next to one another, connecting them electronically. We ultimately abandoned this idea, since it was very difficult with this material to obtain straight edges that exactly lined up.

To facilitate the process, we used **two small wooden slats of 1 cm thickness**, which we attached with **clamps to a table**. The distance between the slats was ***c.* 10 cm**. We also used a **newspaper sheet underneath** to avoid the clay sticking to the surface of the table (a plaster surface works best of all). Before placing the clay between the slats, it is important to prepare it in small quantities. **Each board** requires around **180 g** so we recommend taking something around **220 g** and **kneading it** thoroughly for a minute to get rid of any air bubbles, shaping it into a ball. (If your clay is not even enough, it is better to throw it down forcefully against a flat surface and repeat the process a couple of times.) When you're ready, place it between the slats and gently **press it, bearing down on it** until it is flat enough for the area you want to achieve. We used a **dough roller to flatten** out the clay to 1 cm thickness.

You will notice that the clay is very fragile and less elastic than the industrial type. It tends to split at the edges, which is fine, as long as it is not part of your inner cutter area.



Note: After cutting the hexagonal piece, you might want to smooth the surface with a wet sponge or just using your fingers and adding a little water. This process usually helps to flatten the surface and at the same time avoid any possible cracks.

D – 'Stamping' the circuit

After cutting the hexagon out using the hexagon tile cutter, you can place **the stamp gently down on the clay**, facing the 3D printed side down towards the clay. It is important to apply some force, but quite gently, until the circuit is imprinted in the clay. In this process you force the clay to deform a little at the edges, but you can easily remove the excess material by **sanding it after drying**. Another technique is to place the 'stamp' inside the tile cutter and cut and imprint at the same time. We observed that this also works fine. However, it is difficult to check the quality of the imprint while pressing the cutter down. Also, the circuit tracks will be thinner, which might make the painting process a little more difficult.

E – Drying and sanding

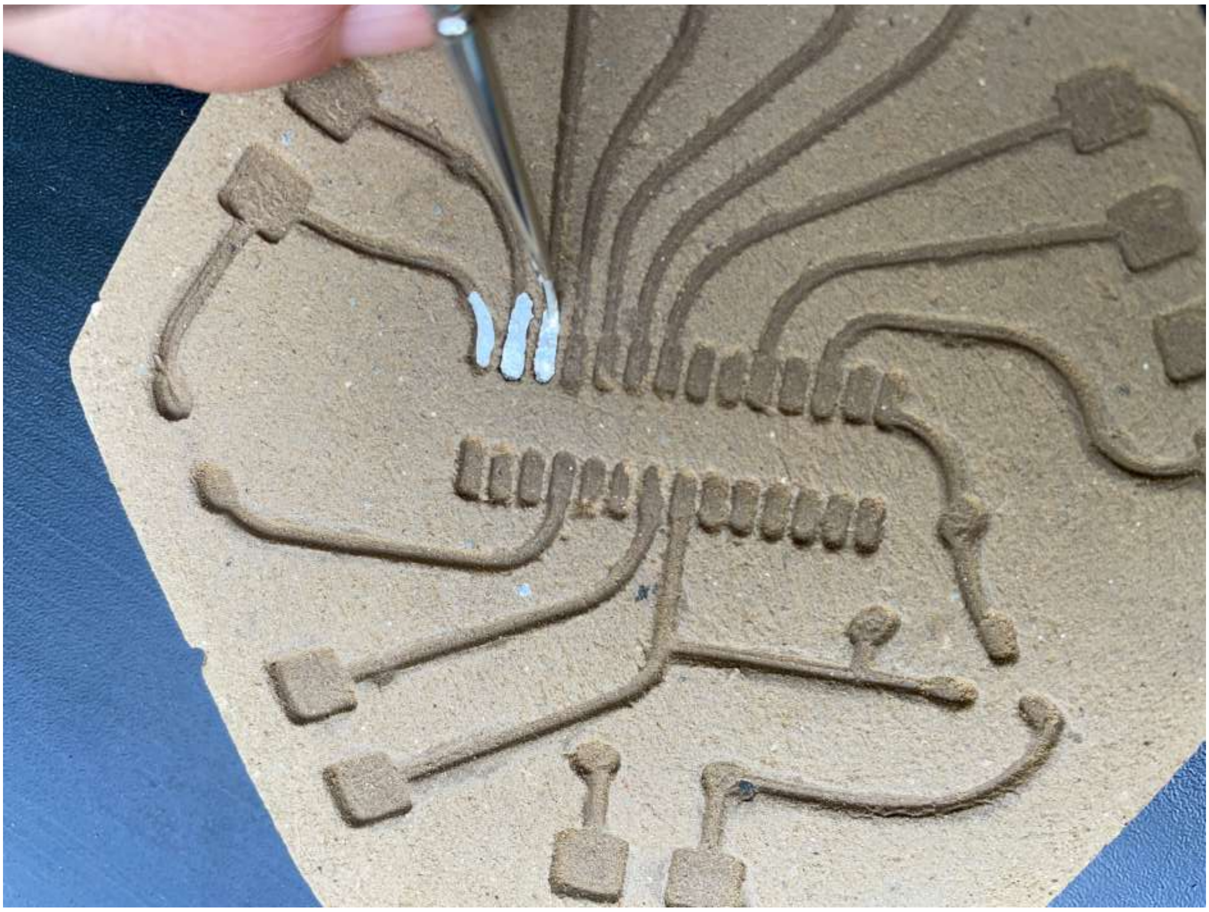
We usually let the boards dry naturally **outdoors for 24 hours** before painting, but this time frame is weather-dependent. If you have more time between modelling and firing the clay, you will ideally **dry the boards indoors for one to two weeks**, positioned between wooden plates and applying some weight to the top plate. In this way they will not deform while drying and will maintain their flat surfaces (we recommend using newspaper between the boards and the wood). If you want to dry the boards quickly, you can also place them **around a wood fire**. It is important to **avoid temperature clashes** during drying so it is best to bring them to the fire as slowly as possible, one step at a time.

You will know when a board is 100% dry when you see its colour becoming lighter and more even. When it is not completely dry, although the edges turn a lighter colour, in the middle the clay is darker and wet. Our experience tells us that this is the minimum drying period required before starting to paint the circuits. If you wish, you can facilitate the painting process by gently **sanding** the boards, using a **120# or finer sanding paper**. After sanding, make sure there is no dust on the board so you can start painting it.

F – Painting the circuit

While searching for **conductive materials** that can be used in ceramics, we came across a **gold lustre** (used often for the gold details on porcelain) that, after firing, becomes conductive. The first problem we encountered was that this product is usually sold by ceramic shops that don't supply any information about its ingredients, especially on the sources of the gold and its commodity chain. The second problem is that it is not possible to solder directly on this gold lustre, so we had to add another precious metal to the equation. The challenge was finding – within the **solderable** and **easily available precious metals**, such as tin, copper, brass and silver – one that could bear the **firing process, which is *c.* 700 ºC,** and at the same time maintain its **conductive properties**. As we know, **tin**, which is mostly used for soldering, melts at a very low temperature, **copper** melts at approximately 1000 ºC, but the oxidation process happens so quickly in the fire that it loses its conductive properties, and the same happens with **brass**. We were left with **silver**, which, although it also oxidises with the fire, maintains its conductive properties. Also, silver is cheaper than gold and widely used by goldsmiths. We were able to find a silver paint, commercialised by a German company, that is made with waste silver powder collected by jewellery makers. It's like an urban mining technique of silver dust.

To paint the circuit, you will need a very thin brush, size 0/5. We recommend starting from the middle, at the place where the ATmega chip will be soldered. These, and the input and output connector pins, are the ones to which you should apply more silver paint. It is important that the lines of silver do not touch one another. If that happens (and it always does), you can correct it using a thin metal piece and scratching it out. For the input and output connector pins, since the paintable area is larger, we used a thicker brush (0 or 1).



There would have been many other ways to print the circuit without the time-consuming hand painting, for instance using a stencil mask and either spraying on it or using another paint-transferring technique. The reason we chose this method is because it appears to be more economical and sustainable, since you generate almost no waste paint.

G – Firing

We fired the boards in our own backyard, reusing a hole that had been dug specifically for the purpose. The wood was collected *in situ,* consisting of dry wood sticks and old branches from our trees. We started a normal fire to generate some heat and placed all the boards around it to complete the drying process. Meanwhile, we collected wood sticks of approximately the same size but in two categories of thickness. The thicker sticks can be used for the base of the boards and the thinner ones can go on top.

While the first fire is settling, you can start to build the 'bed' for the boards by placing the first layer in parallel and the second transversely on top. Using proper fire-proof gloves, glasses and clothes, start to add the boards to the top of the 'bed'/ rack. We used BBQ tongs to handle the boards.

Quickly add the second layer by repeating the same process, only now with thinner sticks. This will cause the fire to expand and also provide an oven effect for the ceramic pieces. The maximum temperature should be around 700 ºC, bearing in mind that it can be hard to control. Our experience tells us that 20 minutes is the average time they need to be ready, so you will need to keep the fire alive during that time. After 20 minutes, you can let the fire go down and check the boards. You should be able to see them glow in the fire, which is when you will know they are ready. Using the tongs, you can quickly transfer them from the fire to a bucket of cold water and leave them there, still holding them with the tongs for a few seconds. This is usually the ultimate 'proof' test for the clay. If there are no air bubbles, stones or cracks, and it has dried properly, it can resist the cold water.

After firing, the silver will turn white as a result of oxidation, but if you polish it well afterwards, it will turn silver again.

Part 3: Programming (how to use the bootloader to upload a program to the ATmega chip)

A – Burn the bootloader onto the ATmega328

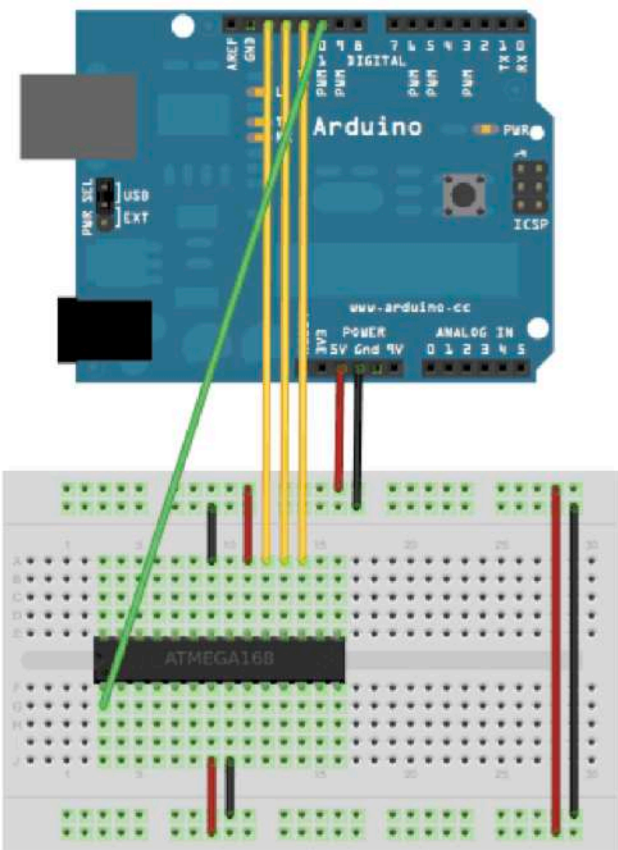
Although the Arduino website has a very good tutorial that explains this step, we will provide a summary of the important steps here. There are two ways to do this, depending on two different scenarios: either you just want to migrate and upcycle your ATmega from an old Arduino Uno board; or you have a single loose ATmega chip (without an attached Arduino board). Our tutorial requires the latter, using a minimal circuit and eliminating the external Arduino clock (avoiding adding a 16 MHz crystal to the circuit) and using the ATmega328 internal 8 MHz RC oscillator as a clock instead. Besides the ATmega328, you will need an Arduino board, a breadboard and some jump wires.

A1 – The first step is to **install support for an additional hardware configuration**. This step is very well documented and explained in the Arduino tutorial, so we will simply copy the instructions from the page here.

1. **Download this hardware** configuration archive: breadboard-1-6-x.zip, Breadboard1-5-x.zip or Breadboard1-0-x.zip, depending on which IDE you are using.
2. Create a **'hardware' sub-folder** in your Arduino sketchbook folder (you can find the necessary location in the Arduino preferences dialogue). If you've previously installed support for an additional hardware configuration, you may already have a 'hardware' folder in your sketchbook. If you don't, add one.
3. Move the breadboard folder from the zip archive to the 'hardware' folder of your Arduino sketchbook.
4. Restart the Arduino software.
5. You should see that it says '**ATmega328 on a breadboard (8 MHz internal clock)**' in the Tools > Board menu.

A2 – The circuit

To use an Arduino board to burn the bootloader onto an ATmega on a breadboard using the 8 MHz RC internal oscillator, follow the diagram below:



A3 – Burning the bootloader

To burn the bootloader, follow the steps that are copied and edited from the [instructions from the Arduino page] after wiring everything as per the diagram before.

1. **Upload the Arduino ISP** sketch onto your Arduino board (you will find it on Arduino > File > Examples > 11 Arduino ISP; select the board and serial port that correspond to your board from the Tools menu).
2. **Select from the Tools** > Board menu, 'ATmega328 on a breadboard (8 MHz internal clock)'.
3. Select '**Arduino as ISP**' from Tools > Programmer.
4. Run Tools > **Burn bootloader.**

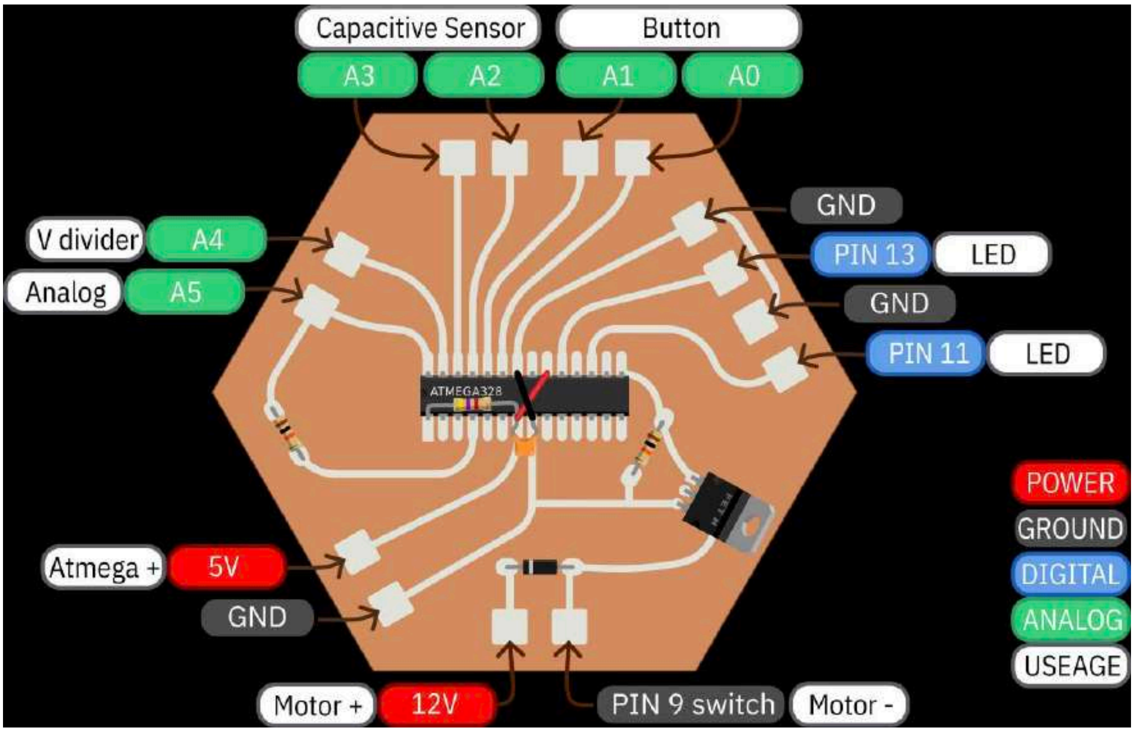
Note: If you encounter errors, please follow this thread for troubleshooting:

https://www.instructables.com/How-to-Fix-Expected-Signature-for-ATMEGA328P-Is-1E/

B – Uploading the program

B1 – Understanding the program

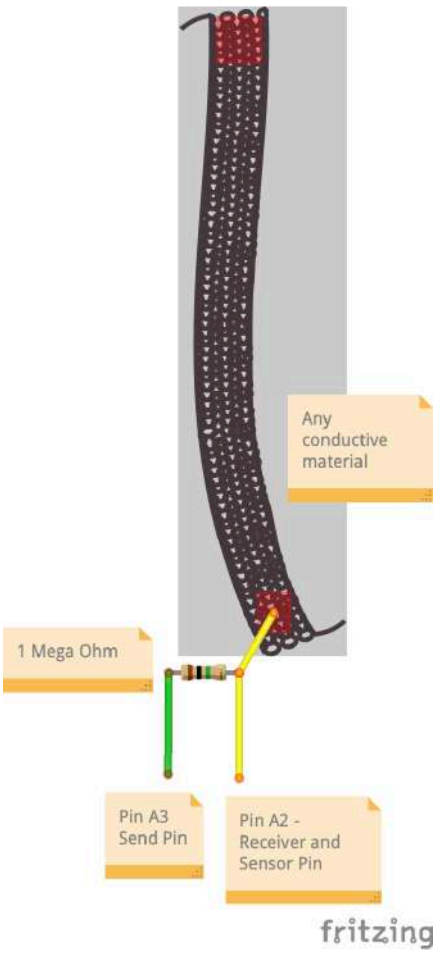
Our program was designed to recognise the inputs on the board and to automatically use the signal in a determined output pin.



The digital **input 'Button' (pins A1 + A0)** is switched ON whenever the circuit between the two inputs is closed. Clicking on the button will immediately affect the digital outputs (**pins 13 and 11**). To these pins, we can attach an **LED or a piezo buzzer** (a DIY speaker could also work but would require more electronics to amplify the signal). This means that if I **click on the button** repeatedly using a certain rhythm, it will immediately control the pitch of the piezo buzzer's sound (according to NOTES previously defined in the program) or by flickering the LED. The sound pitch and flickering rhythm can be 'distorted' by the **capacitive sensor (pins A3 + A2).**

To program the capacitive sensor, we used the famous Arduino **CapacitiveSensor.h Library**. (to run the program in your Arduino IDE, you will need to follow this tutorial: https://www.arduino.cc/reference/en/libraries/capacitivesensor/).

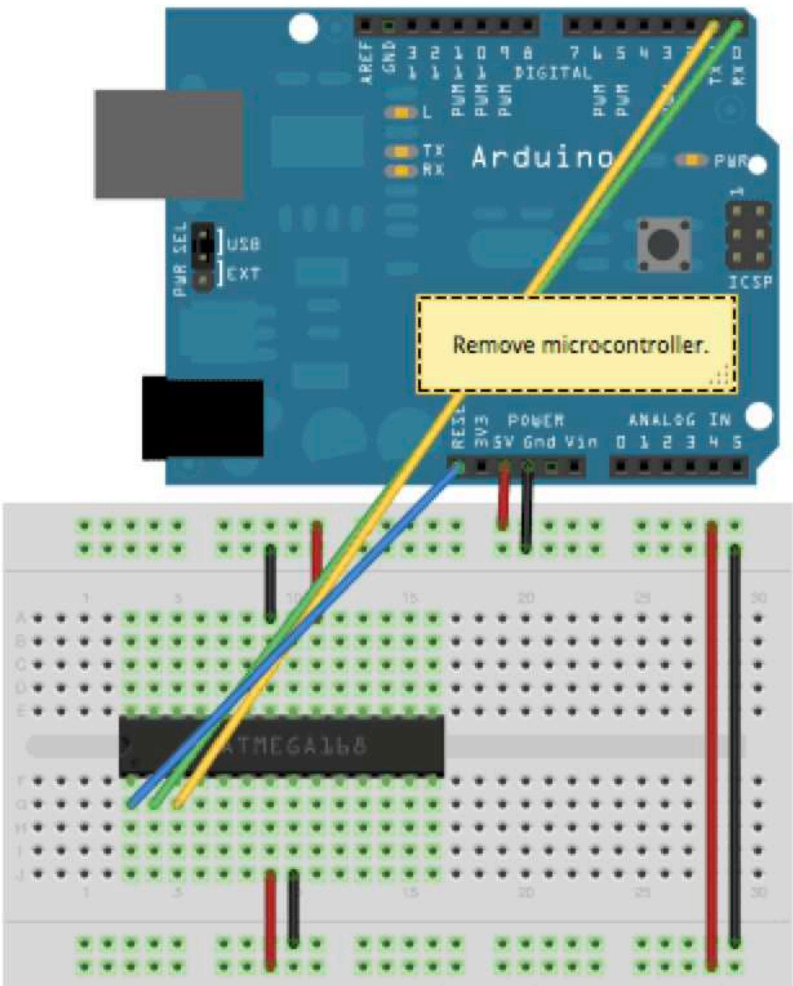
A2 is our sensor pin (or receive pin) and **A3 our send pin**. Just place a high resistor (at least 1 Mega Ohm) between the two pins and attach a **conductive material to sensor pin A2.** You can use any conductive material to create a capacitive sensor (any metal, copper tape, e-textile).



When you click the button for more than two seconds, it automatically switches to the motor output. A **12 volt DC motor** can be directly connected to **pin 9**, but you will need to power it separately with a different power source (we will cover this part later in the tutorial). Interacting with the **analogue A5 pin** and **voltage divider A4** will affect the **speed of the motor**. **Note:** You can use any analogue variable resistive sensor (a normal light sensor LDR or even a piece of fruit) directly connected to pins A4 and A5, but do consider that you might have to replace the resistor on the board for better results, depending on the resistance of your sensor.

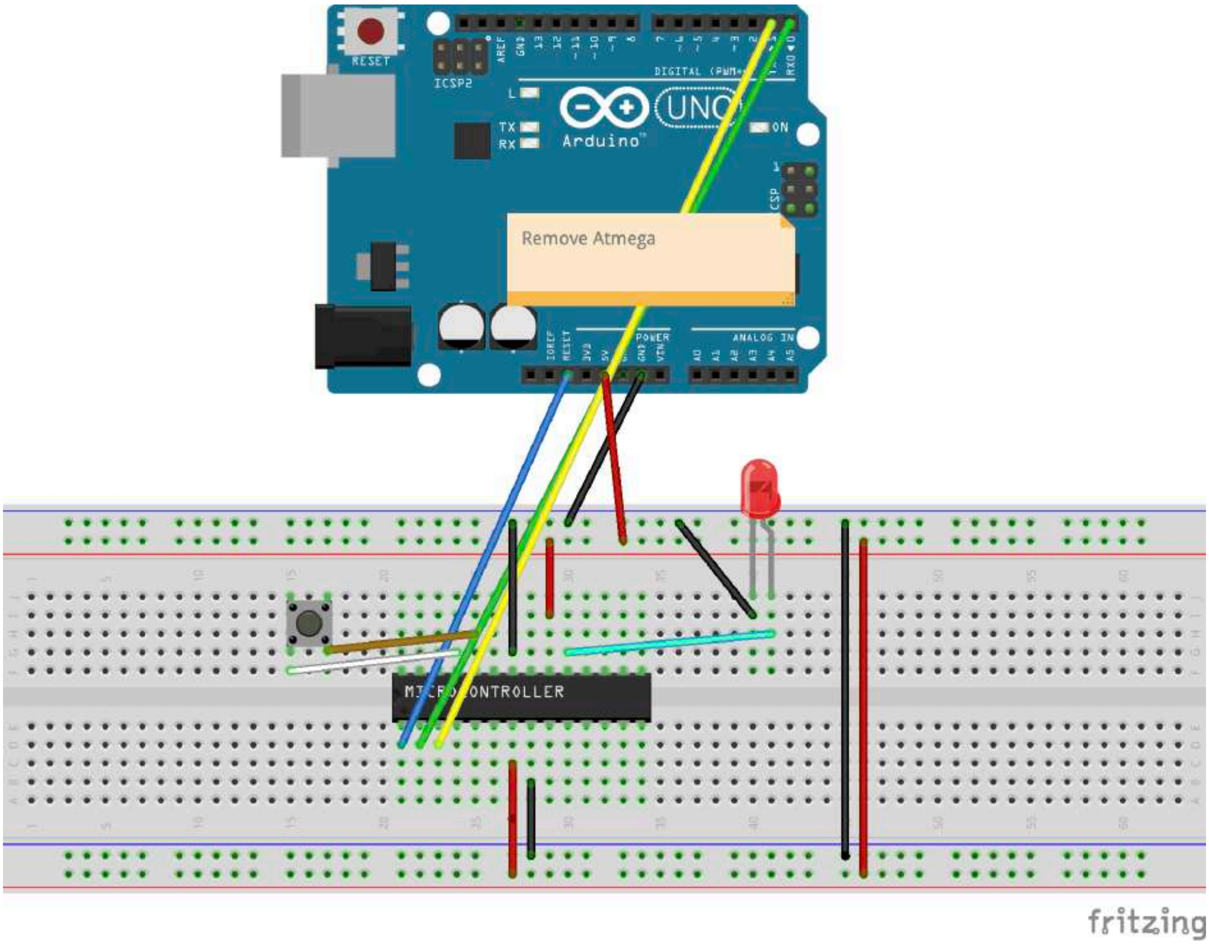
B2 – Uploading the sketch to the ATmega

Follow the schematics to wire your ATmega on a breadboard to an Arduino board. IMPORTANT: **Remove the ATmega from your Arduino Uno**. With this method, we can use the Arduino board as an interface to communicate between our ATmega and the Arduino IDE. The **RESET pin, TX, RX, 5V and GND** of your **ATmega** need to be connected to the same pins on the Arduino board. Open the sketch in the Arduino IDE, and before uploading make sure that you select from the Tools > Board menu**, '****ATmega328 on a breadboard (8 MHz internal clock)'.**



B3 – Quick test

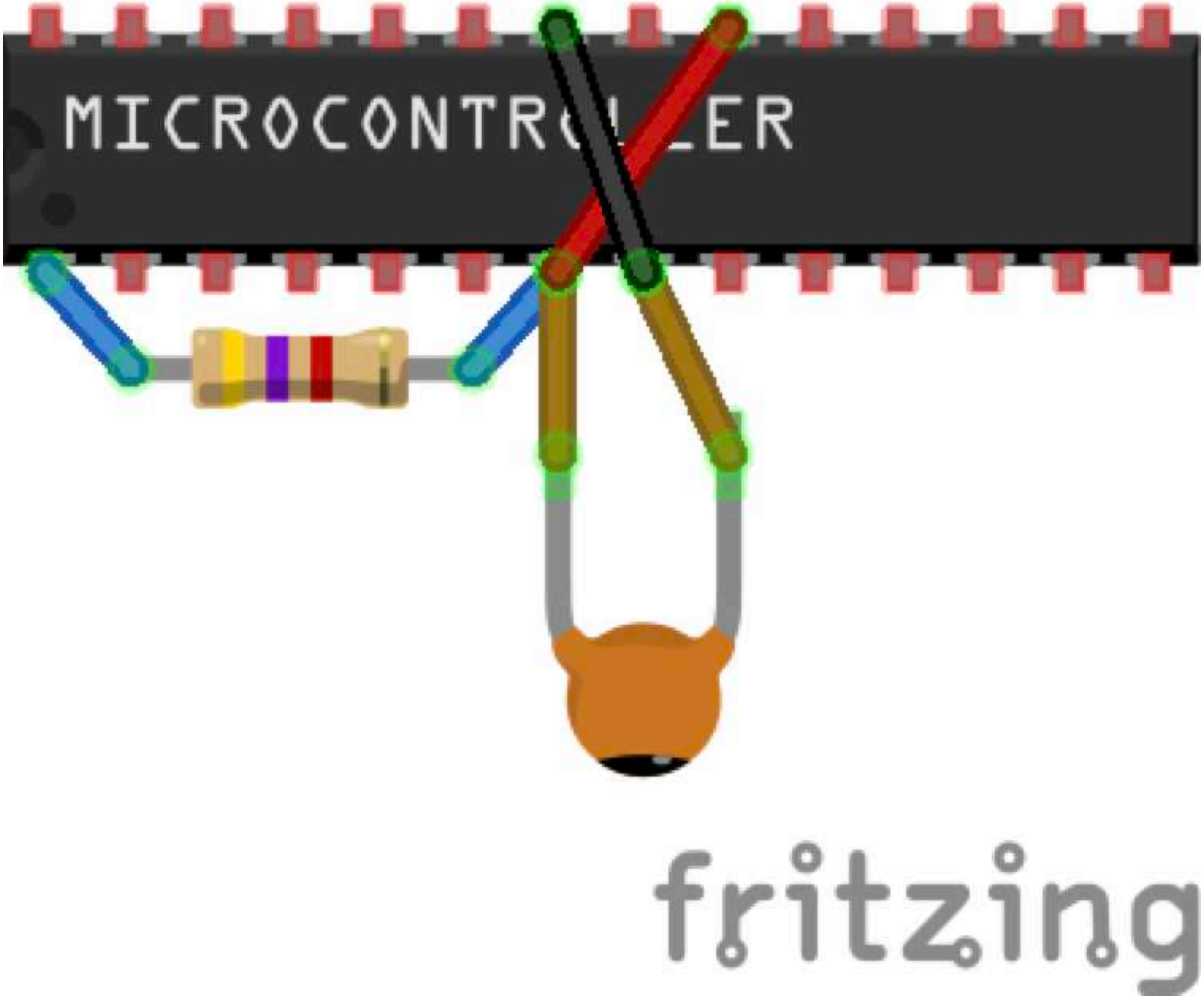
You can quickly test if the program is working by connecting **buttons A1 and A2** and an **LED** (light emitting diode) to **pin 13 or pin 11**. If the LED flickering reacts to the rhythm of clicking the button, your program should work.



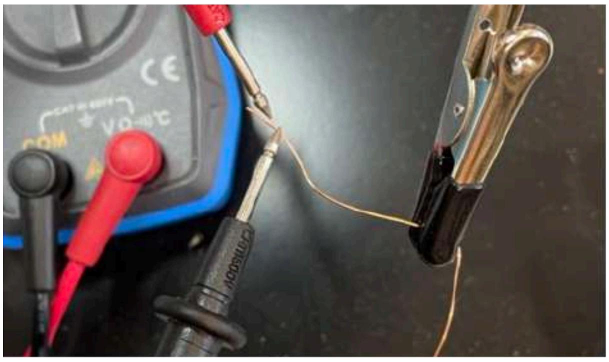
PART 4: Electronics (soldering the electronic components)

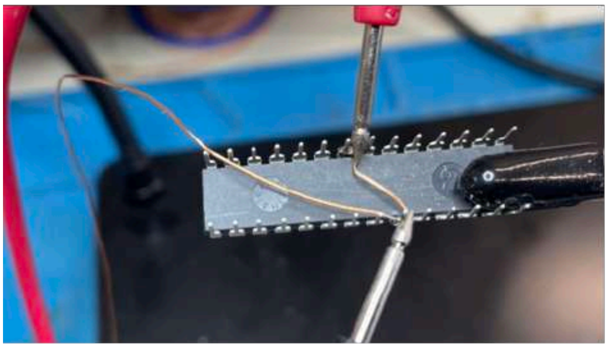
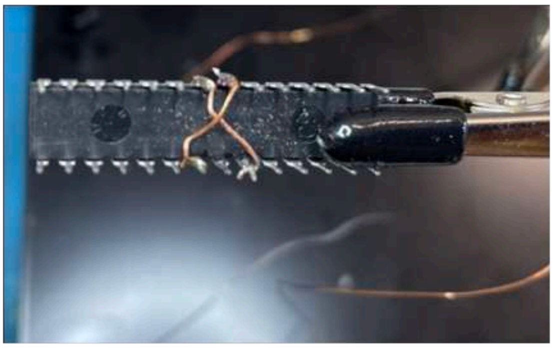
A – Preparing the ATmega

Before soldering the components to the board, we prepared some of the connections directly on the ATmega. We had to **bridge across the chip the VVC pin and ground**, and add a **4.7 K resistor between the RESET pin and VVC**, and a **100 nf ceramic capacitor between VVC and ground** (we opted for this step to avoid crosses in our single-layer printed circuit).

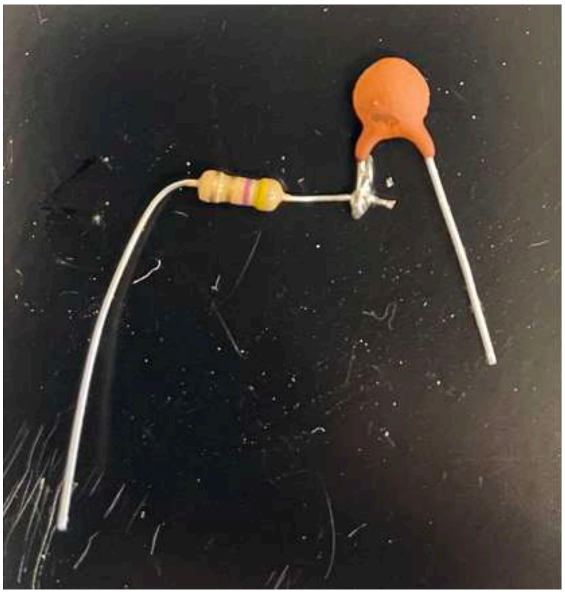
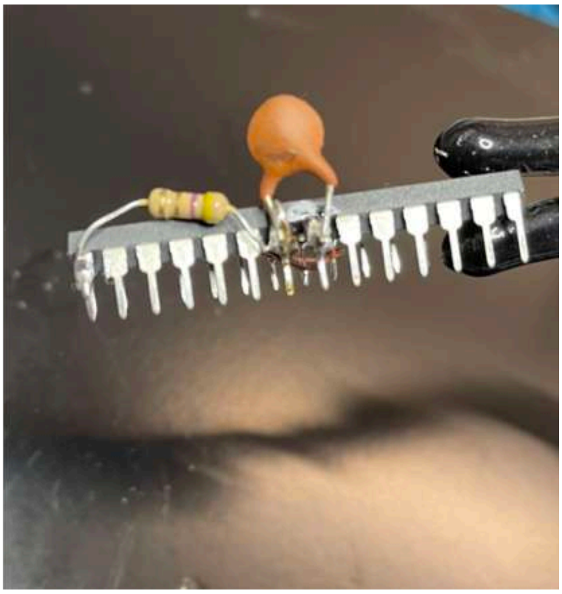


To avoid having several wires on top of the ATmega, we used a very thin coated copper cable (in this case we had one in our workshop) to connect **VVC to VVC** and **GRD to GRD**. The wire we used was insulated to make sure that the **wires could not touch one another,** causing a short circuit. Before soldering it to the right pin, it is necessary to sand out a little of this coating material at the edges where you want electrical contact. Before soldering it to the chip, **test for conductivity** using a multimeter and repeat the process after soldering.



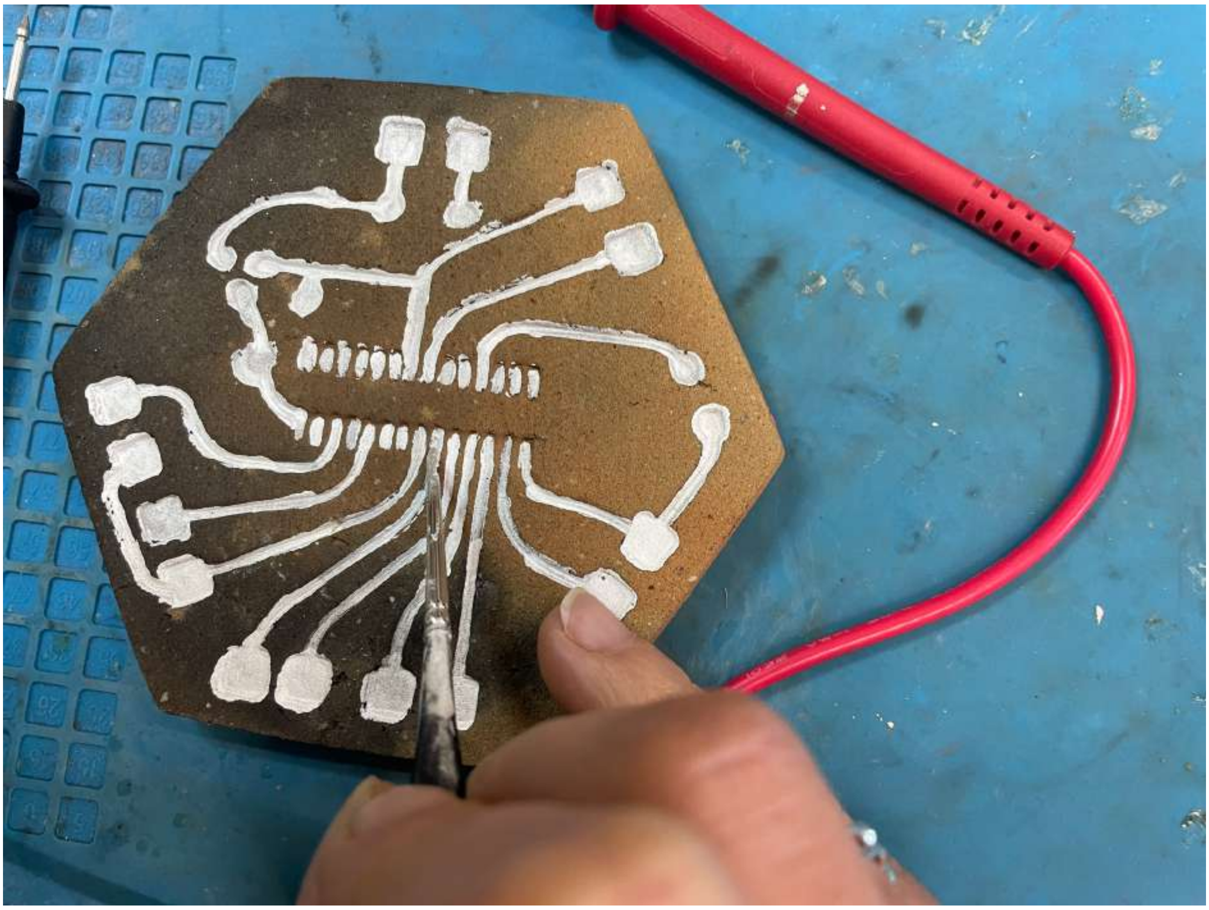



We joined the **capacitor with the resistor** before adding them to the ATmega.

B – Adding components to the board

B1 – First, we **checked conductivity** using a normal **multimeter**. If you find small parts that are not conductive, you can add a little silver paint on top to correct them.



B2 – All our **parts are recycled from old electronics.** Here is a list of what you will need:

- **Resistor** 10 K (x 2)
- **Schottky diode**
- **Power Mosfet N-Channel** to power the 12V motor (we used the IRLB 8721, but a similar component can be used as long it has the same wiring diagram – see below)

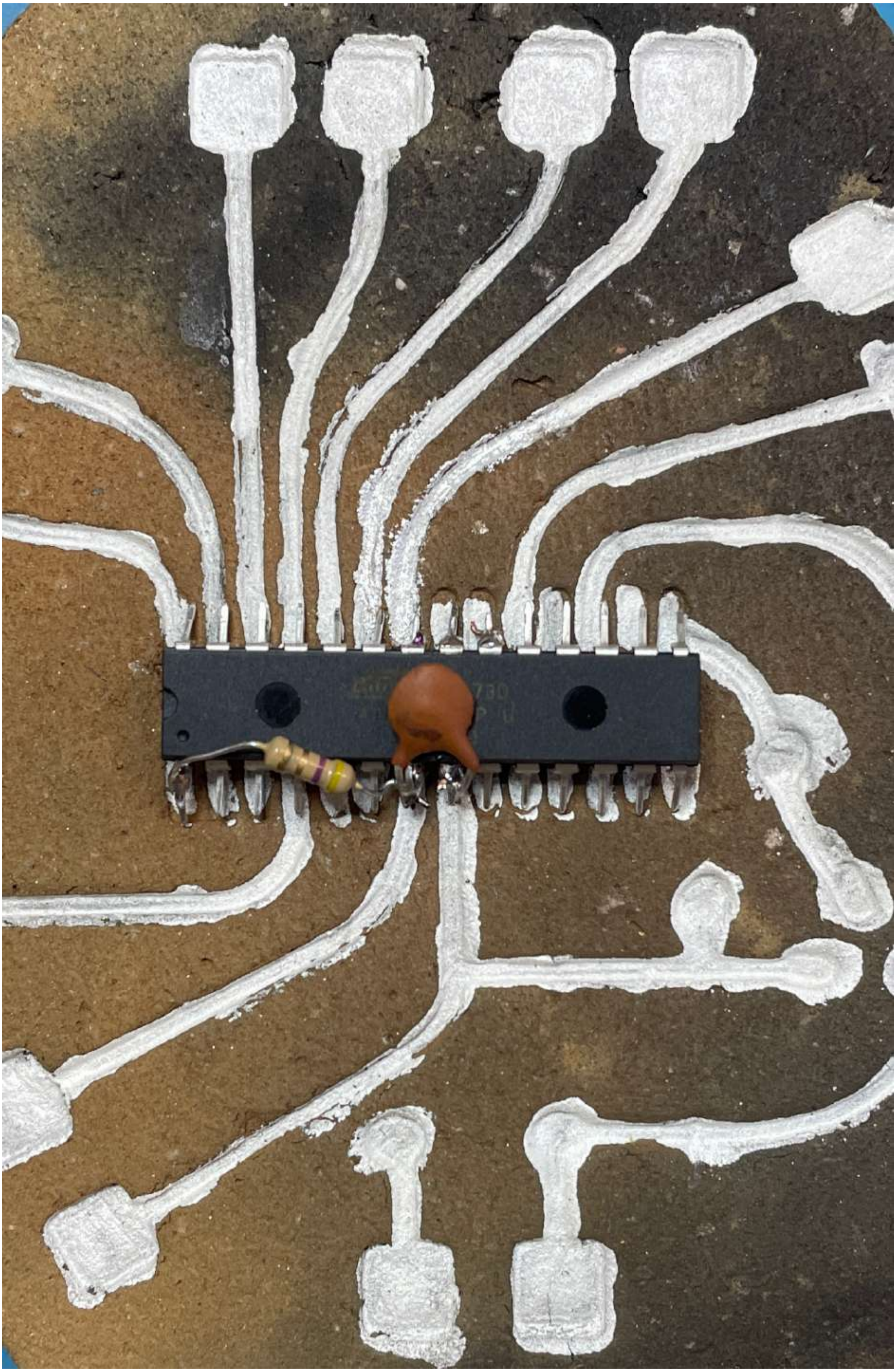
G	D	S
Gate	Drain	Source

B3 – Soldering

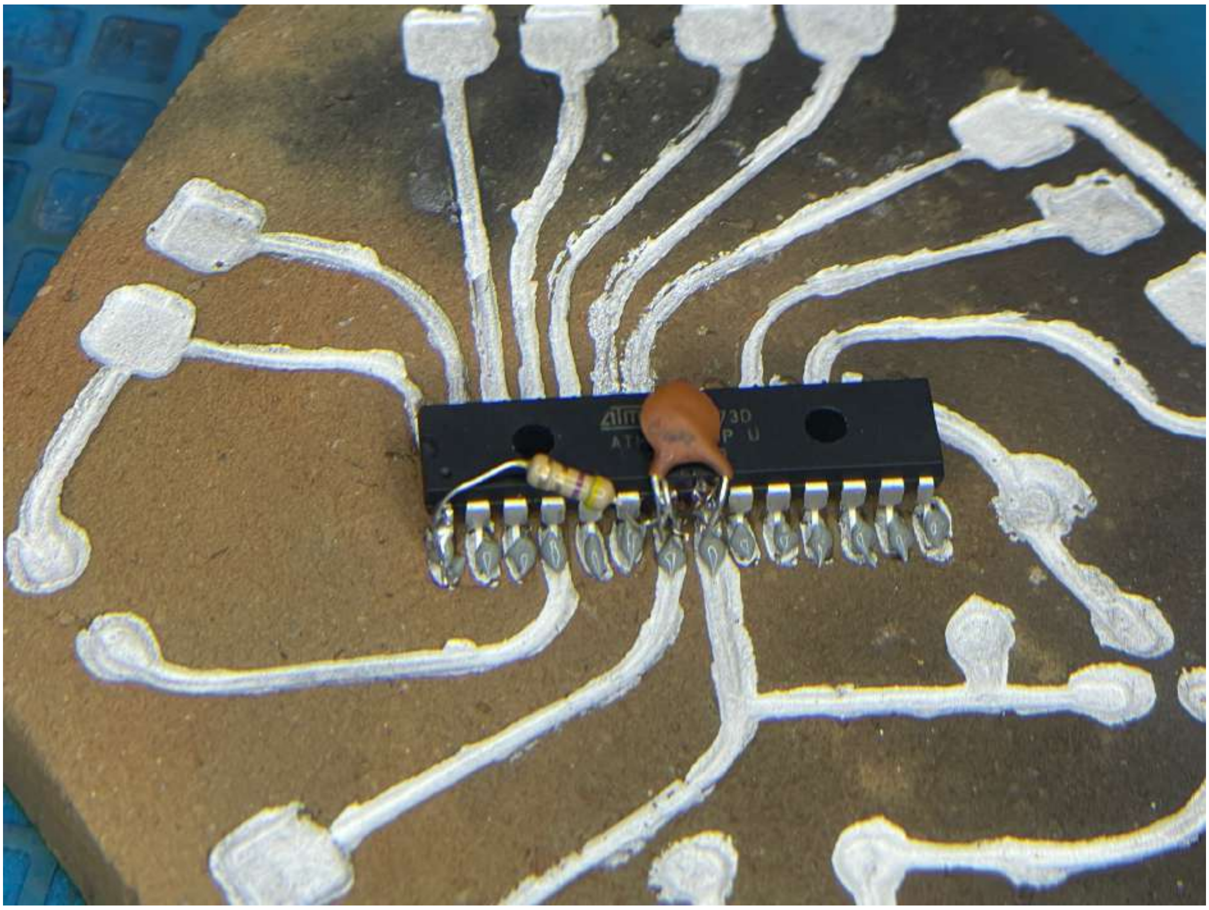
Because the silver connectors on the ceramic PCB are very sensitive, we concluded that the **SMD soldering** method with **hot-air soldering** would be more appropriate. We used a **lead-free SMD soldering paste** ISO-Cream ® 'Clear' from Felder, a German brand that follows fair-trade standards, thereby meeting our demands for ethical hardware.

Before soldering, ensure that your working area is ventilated and free of flammable materials. We recommend using a fume extractor and a mask during the entire process.

We **started with the ATmega** by bending the connectors in a way that fitted our own board (all ceramic PCBs are unique pieces, so you will need to find the best way to fit them on top of your board). **Note:** Always double check the direction of the ATmega in the circuit.



After adjusting the connectors, you can start to **gently apply the soldering paste** to each of the ATmega pins.

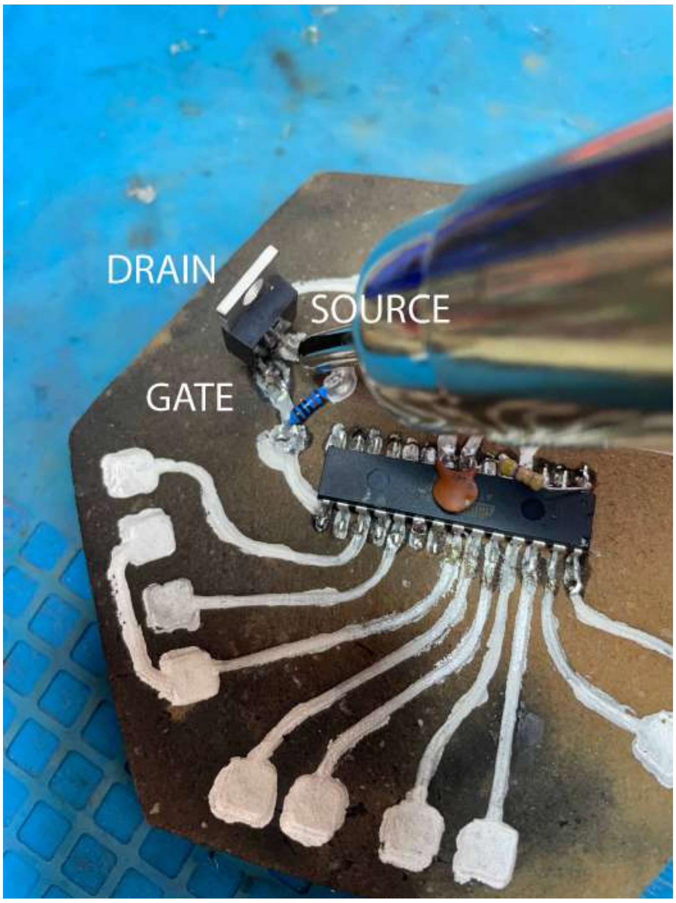


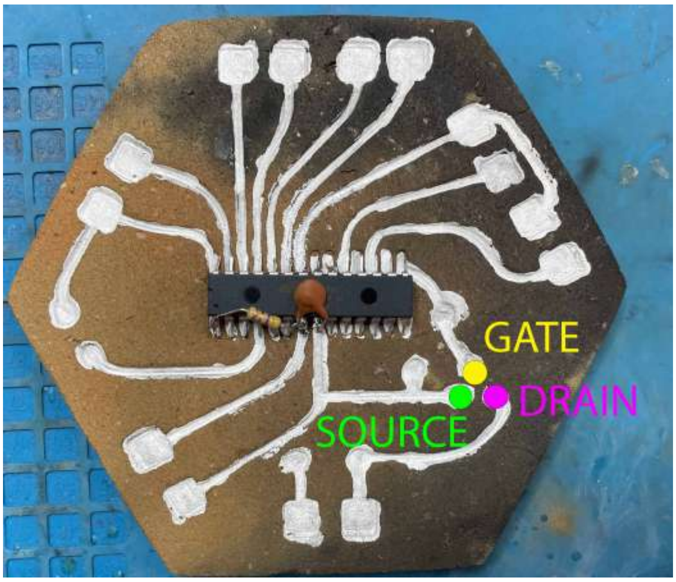
Make sure that the soldering paste is not touching two connectors. After verifying all of the pins, you can start to hot solder it. Pay special attention when using hot-air soldering; since we don’t see the heat, we tend to pass our hands and tools over it. Be especially careful in this process and do everything slowly and patiently. When you’re ready, bring the soldering tip close enough to the paste and wait a few seconds until it melts and turns silver and glossy.

Repeat the same process with the resistors and the diode.

Before soldering the **Mosfet,** verify the schematics, using the ATmega position as a reference:

- Gate (left) and source (right) are attached on the front in the same order.
- Drain (middle) is attached on the back.

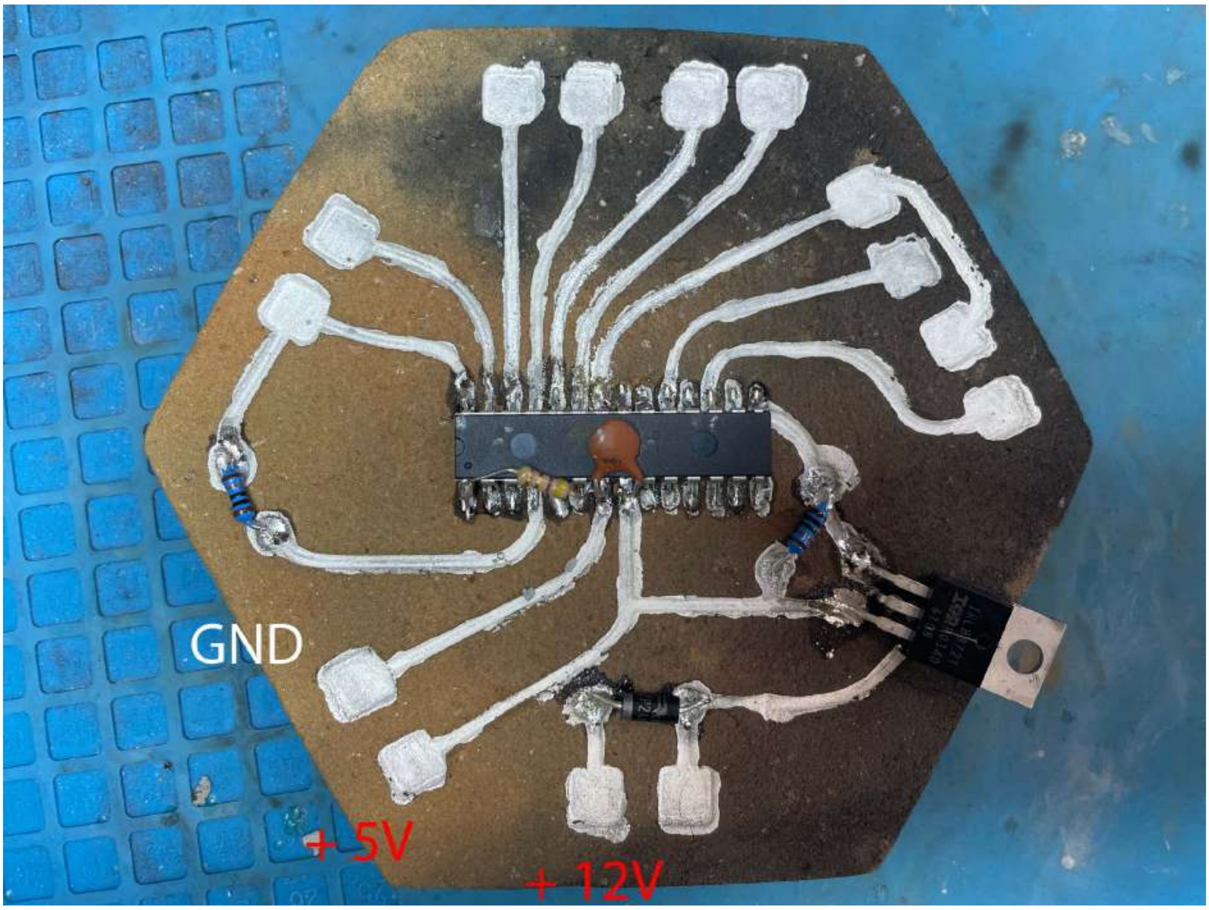




PART 5: Powering and testing

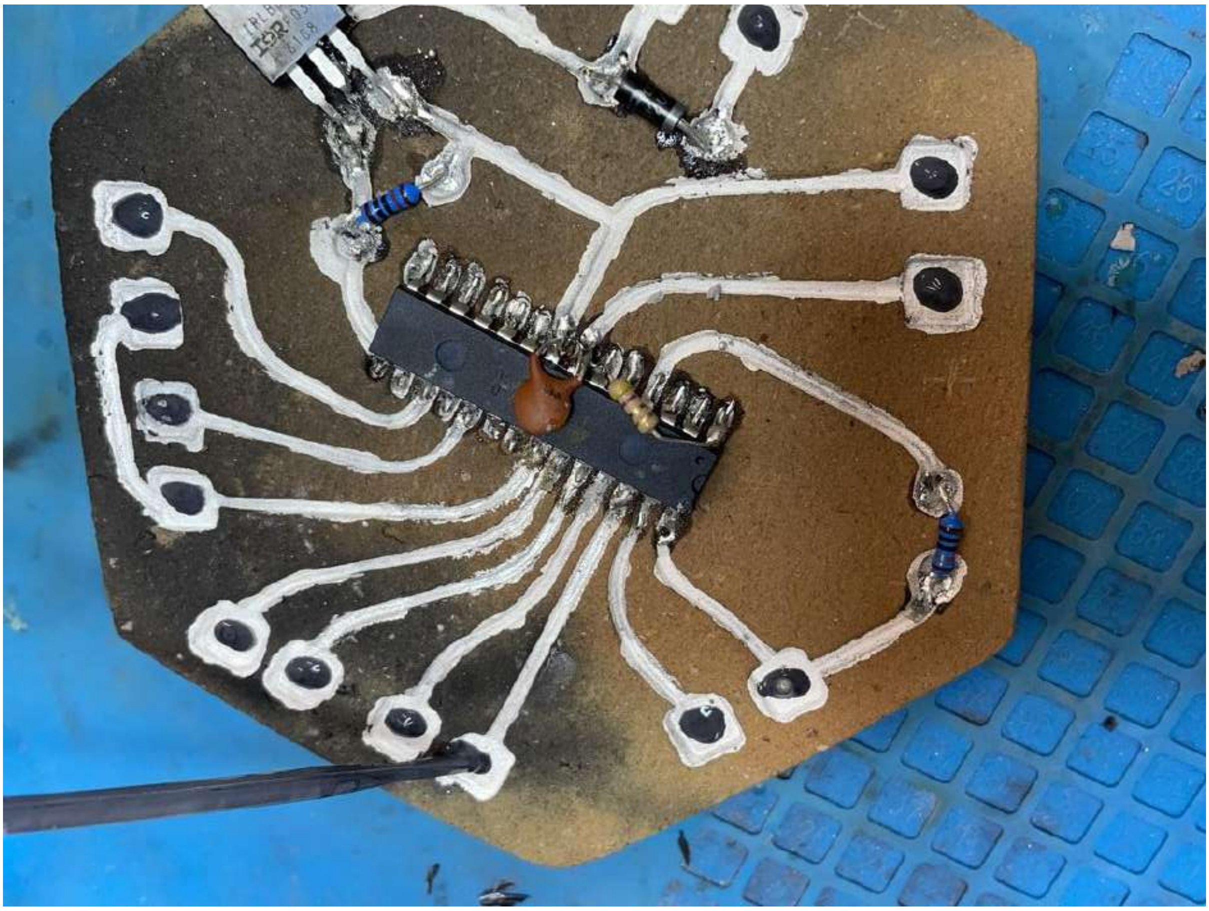
A – To test the board, you will need two power connections:
POWER 1 ATmega – 5V DC
POWER 2 Motor – 12V DC

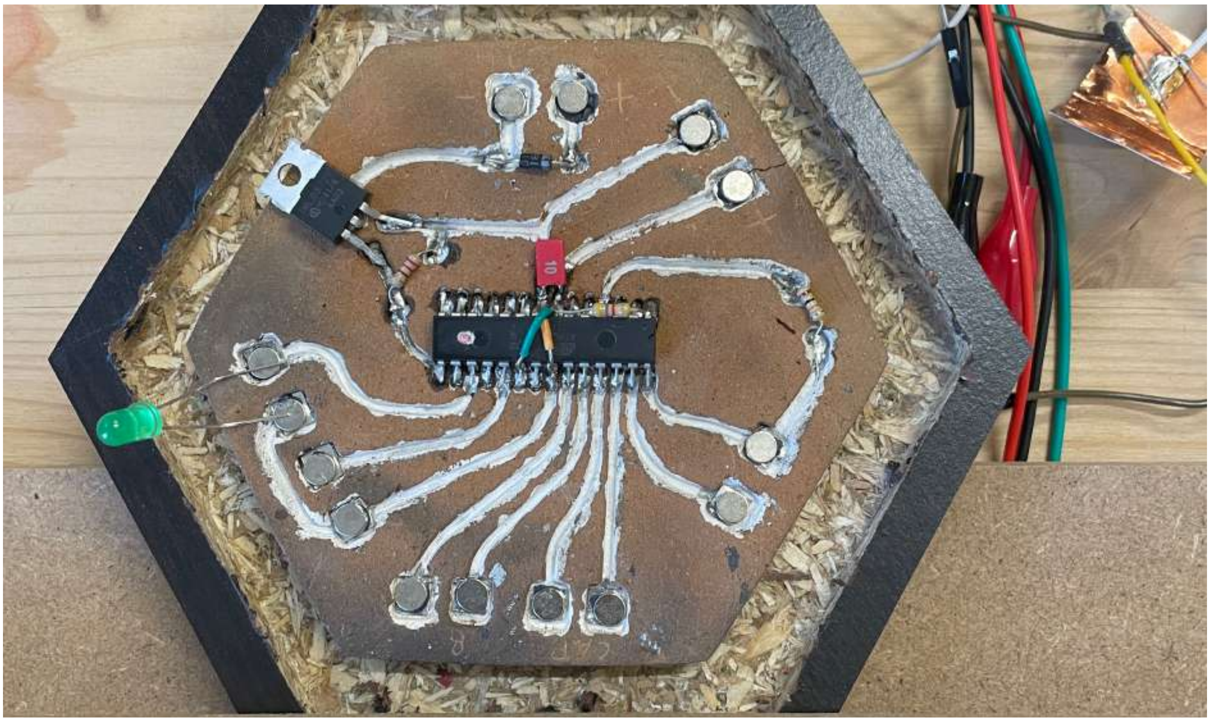
Follow the picture to correctly connect the power.



To facilitate testing, prototyping and playing with the board, we attached **magnets** to each input and output. The magnets were recycled from different sources, but since it is not possible to solder magnets, we had to find another solution. We could not find a fairly traded or ethically responsible solution and ended up using a commercial conductive glue. For this process, you will only need a very small amount of this product. Make sure that you leave it to dry overnight.

Connect your input and output and you're ready to go!





Patrícia J. Reis and Stefanie Wuschitz, October 2023