FEMINIST HARDWARE: MAKING PRINTED CIRCUIT BOARDS WITH NATURAL CLAY BY PATRÍCIA J. REIS & 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 develop and speculate upon renewable practices for the benefit of both nature and humans.
We are exploring different materials, sentient, low-impact, non-toxic, fair 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.

This tutorial aims to provide a step-by-step manual to share the experiences acquired through practice based research in the process of making Printed Circuit Boards. This tutorial focuses on building PCBs with natural clay using recycled silver as main electrical conductors. Our PCB is in itself 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, but was intended to encourage you to play with it yourself, with the option to modify it or write even, if you feel like it. You can accomplish this by downloading our code from Github, editing and uploading it via the Arduino IDE. Our project is all open sourced based and we are happy to receive feedback and hear about your own experience.

This tutorial is structured in **five parts**, that each contain several steps:
Part 1: Circuit and the 3D print "stamp"
Part 2: Clay (from collecting soil to modeling the boards, painting the circuit and firing it in open fire.
Part 3: Programming (How to use the Bootloader for uploading 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 print "stamp"

Our initial idea was to develop a microcontroller PCB that could work with the chip **ATmega328P**, commonly used in the famous Arduino Uno board (or **Arduina** board how some feminists call it). Why this chip? Because we are part of community Hackerspace — Mz* Baltazar's Lab (a feminist hacklab and artist run off-space based in Vienna, Austria)— and the Arduino Uno has been our favorite microcontroller in the past 12 years. After using it in many prototypes, artworks, workshops, we had many malfunction Arduino boards left. But it turned out that their chips were actually still working. The idea was to re-use these chips in our new project.

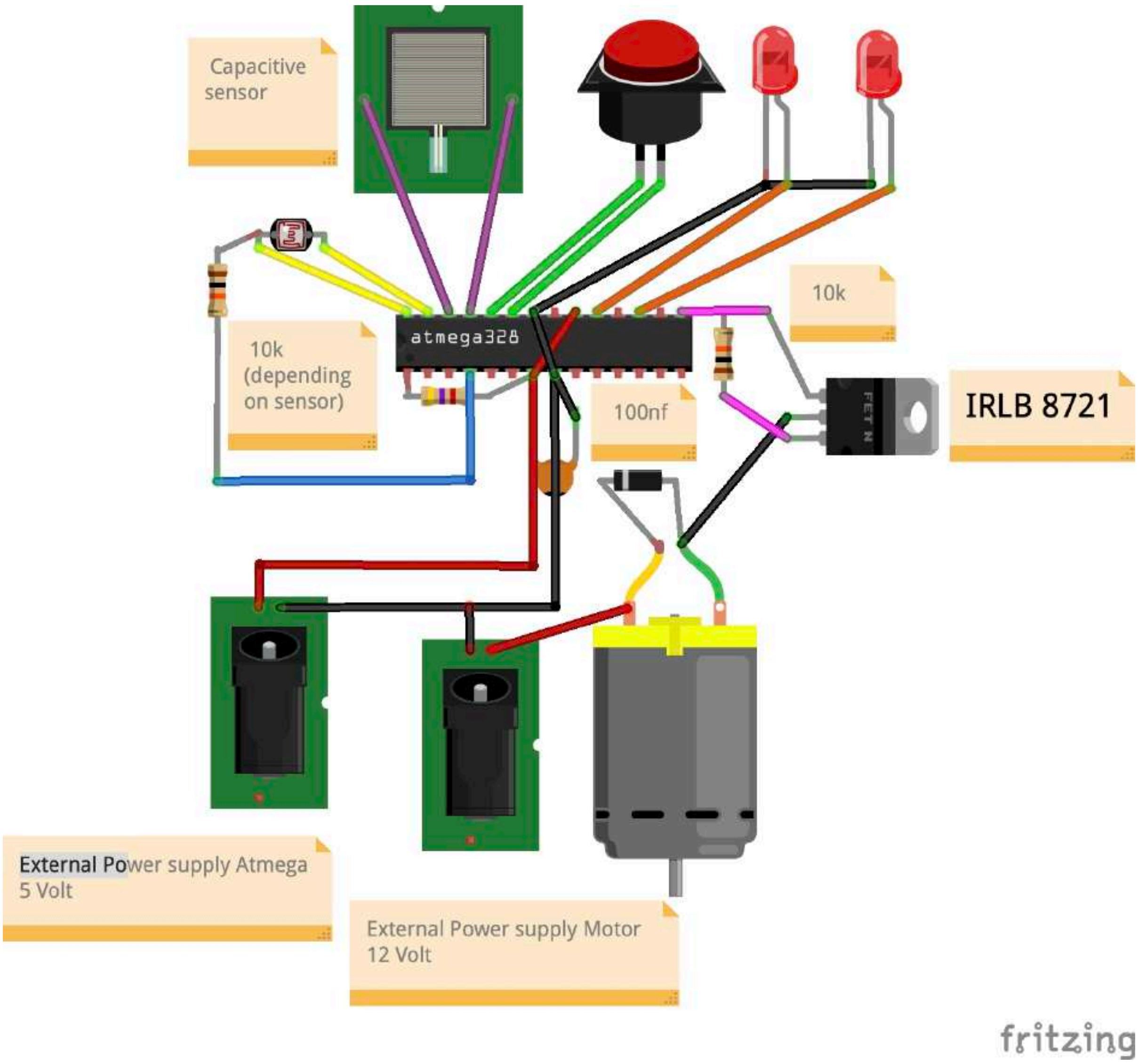
The second thought and challenge was to come up with an electric circuit that would allow us to receive several forms of **input signal** (analog and digital sensors) and generate a variety of **output signal** (to control leds, motors and speakers). Using the lowest amount of the ATMEGA chip pins possible, in order to simplify the circuit, we drew a diagram with Fritzing for the following functions:

Input:

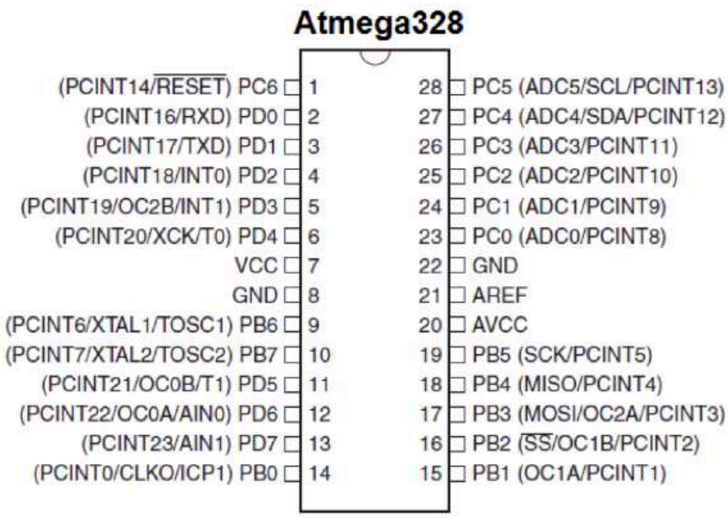
- Capacitive sensor - Atmega Pins A3 + A2
- Analog sensor with a voltage divider - Atmega Pins A4 + A5
- Digital button - Atmega Pins A1 + A2 + 2

Output:

- 2 x leds or piezo buzzers - Atmega Pins 13 + 11
- DC motor with a transistor circuit - Atmega Pin 9



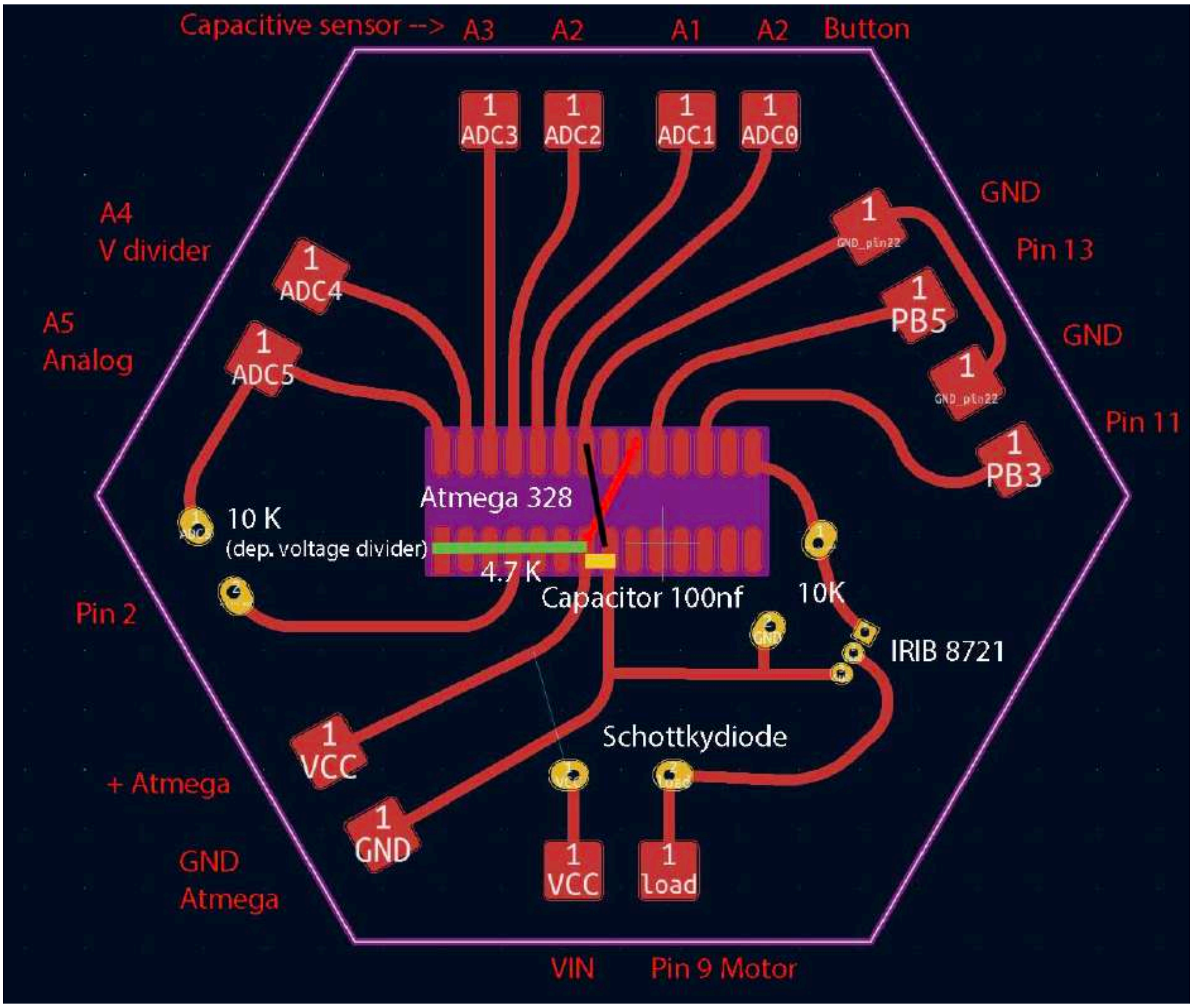
Here s schematic to help to understand the **Atmega328** pinout:



We will follow up on the details concerning the electronic components further down in this manual in the Part 4: Electronics (Soldering the electronic components).

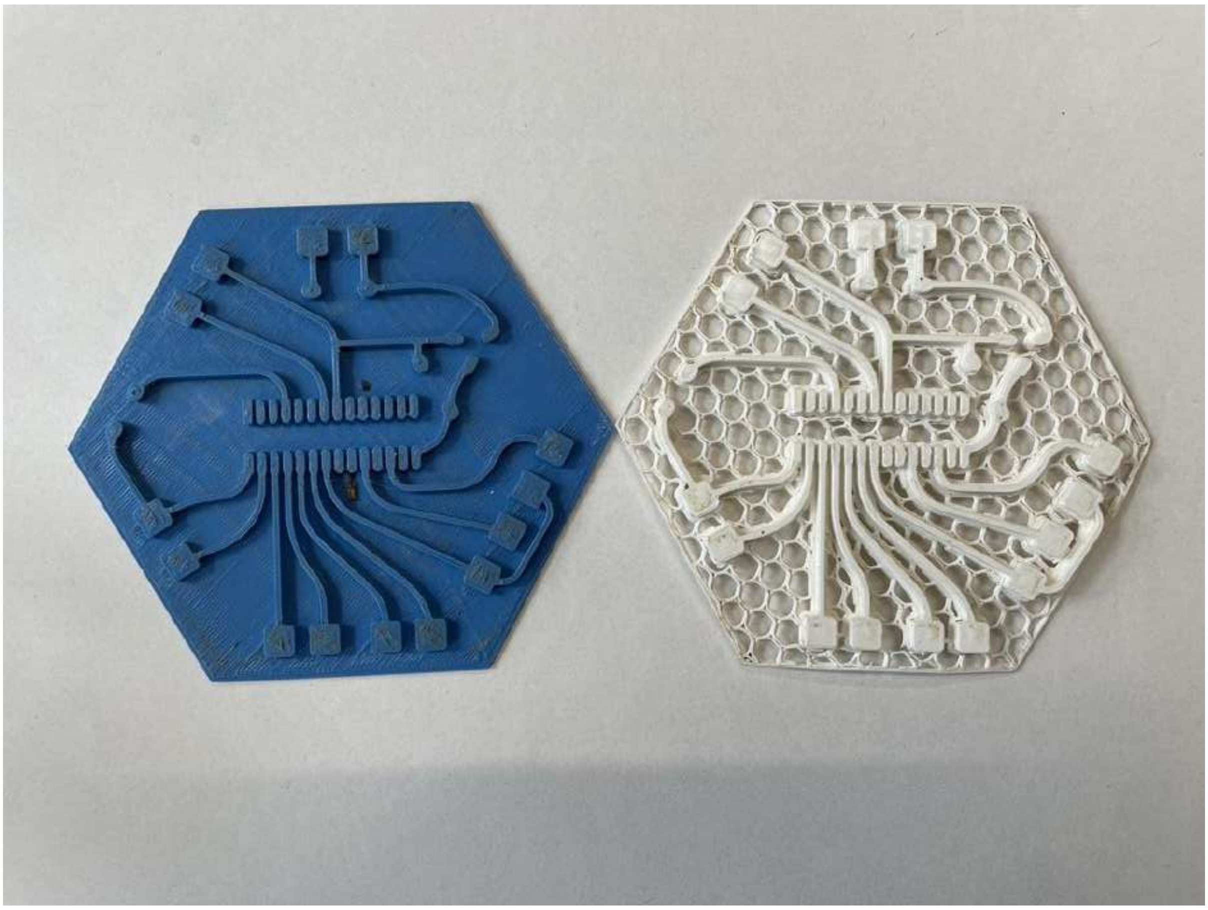
The next step was to **simplify the circuit** in order to reduce it to one **single printable layer**. To avoid multiple layers, we decided to solder some of the connections and electronic parts directly on top of the **Atmega328** (explained also in detail later). These are connecting the **VCC pin 7** with **AVCC pin 20**, as well as **Ground pin 8** to **Ground pin 22**. We also added a bridge from the **RESET pin to VCC** through a 4.7K resistor.

Here a diagram of the circuit in an hexagon shape:



With this insight we were able to design a **3D printed "Stamp"** using recycled Polypropylene filament. This process took a while, because it had to take into consideration the clay's changed size after drying and firing. All clay is shrinking in size considerably during the drying and firing process. We come up with **5% shrinking** percentage, but this number might vary depending on the clay you work with. We also made some experiments with the deepness of the circuit tracks and realized that ideally the imprint should be **1.2 mm deep**.

Here are sample pictures of some of the “stamps” and the files for **3D printing** it yourself can be found here.



PART 2 - Clay

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

To build the **base of our PCB** we needed insulating, sustainable and robust materials (maybe eggshells? Wood plates? wax? Ceramics?). We immediately went for ceramics, in particular **porcelain**, as it already plays an important role in electronic components such as capacitors, Piezo, resistors, etc… Porcelain is an industrial made material composed by **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 prospected and mined around the world, in small scale in Europe and bigger scale in China, Brazil, South Africa, Vietnam (among others). In pottery, Porcelain, also known as **China Clay**, is a very delicate and sensitive material, (we could say it comes with its own agency), more difficult to control if compared with other industrial clays. Also, along with the other harder and resistant stoneware clays it usually requires higher firing temperatures in two Stages: a first firing known as ceramic **bisque of c.a. 1000º C** and a **Glazing firing around 1200º C** in an electric Kiln. During our first experiments with Porcelain we were immediately aware that those higher temperatures and therefore electric consumptions were not compatible with our standards for **Ethical Hardware**.

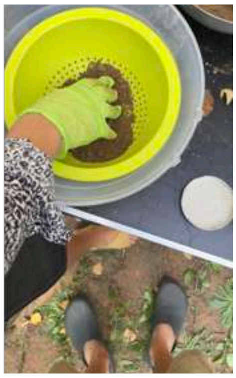
It was exactly when we struggled with the question of how to manufacture clay in low energy and low impact ways, that we came across the work of **Heinz Lackinger**. Heinz Lackinger is a pottery crafter in Donnerskirchen, Burgenland in Austria that works with prehistoric techniques of firing clay in an open wood fire. Instead of sophisticated machines, he only uses a simple hole in the ground of his 18th century backyard house. We had the privilege of spending two days with this skilled craftsman, learning how to identify and collect the clay, how to model it and fire it just using old, dry branches collected from forest ground. If the clay is collected in 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 our own experiments with later applying this technique in the making of Natural Clay PCB boards.

A - Collecting the Clay

As we are walking through the forest on an early fall morning the air is slightly humid. We are a small group of participants and Heinz is leading us 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 color mutates from dark brown to reddish brown and while massaged between the fingers one can sense a certain increase in plasticity. It is definitely clay! We have got **no lab tools** in the forest, only our human **sensory system**. Heinz explains that all landscapes are different and therefore identifying clay in Burgenland might be different than in e.g. Brazil. Nevertheless, a soil with argil properties is finer than sand, different then 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, some with more or less argil properties. We observed that the clay that has got less argil properties tends to shrink more and is not as resistant and water proof.

B - Cleaning the clay

We collected our clay in the beginning of fall in dry weather. The soil is mainly dry but does not consist only in argil: you will find little stones, plants, and even small insects. When the clay is that dry, the easiest way to clean it is by using a net that retains the undesirable waste. Our favorite tool is a normal **kitchen colander**. The waste collected should be given back to earth, back into the ground.



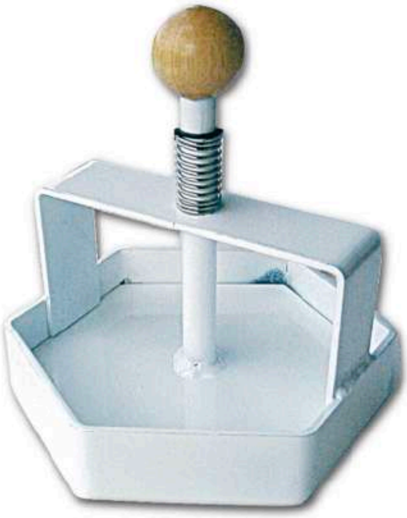
You will end up with a fine powder that will require some added water. We calculated an average of **100 ml of water per 1 kg** of fine powder. Mixing it is just like blending flour and water, but without the inconvenient grumps. You should end up with something like a ball of clay after 10 minutes of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

C - Modelling the clay

For the shape of our PCB board we used a **hexagon tile cutter with 10 x 10 cm** that can be purchased in common [ceramic shops](). The hexagon shape, as the tile form is not mandatory; you can do **any shape** and pick any thickness you desire, as long as it maintains a printable surface of c.a. 10 x 10 cm. We choose this shape and format in hope of assembling the boards as tiles next to each other, connecting them electronically. In the end we abandoned this idea since it was very difficult with this material, to obtain straight edges that could exactly meet each other sidewise.

To facilitate the process, we used **two small wooden slats with 1 cm thickness**, attached with **clamps on a table**. The distance between the slats is c.a. 10 cm. We also used a **newspaper sheet underneath** to avoid, that the clay sticks to the surface of the table. (Ideally a plaster surface works better.) Before placing the clay between the slats it is important to prepare it in small quantities. **Each board** requires around **180 gr** so we recommend to take something around **220 gr** and **knead it** thoroughly for a minute to get rid of air bubbles, to form it in the shape of a ball. (If your clay is not homogenous enough, better to throw it down forcefully against a flat surface and repeat the process a couple of times). When you're ready, place it in between the slates and gently **press it bearing down on it** until it is flattened enough for the area you would like to achieve. We used a **dough roller to flatten** out the clay to 1 cm of thickness.

You will notice that the clay is very fragile and not as elastic as the industrial type. It will tend to break on the edges, which is fine, as long as that part is out of your inner cutter area.



Note: After cutting the hexagon piece, you might want to smooth the surface with a wet sponge of just your fingers and add a little water. This process usually helps to flatten the surface and at the space time avoid any possible cracks.

D - “Stamping” the circuit

After cutting out the hexagon with the hexagon tile cutter you can now place **the stamp gently on the clay**, facing down the 3D printed side towards the clay. It is important to apply some force but only quite gently, until the circuit is imprinted in the clay. In this process you force the clay to deform a little on the edges, but you can easily get rid of that 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, one can’t really check the quality of the imprint, while pressing down the cutter. Also, the circuit tracks will be thinner, which might make the painting process a little bit more difficult.

E - Drying & Sanding

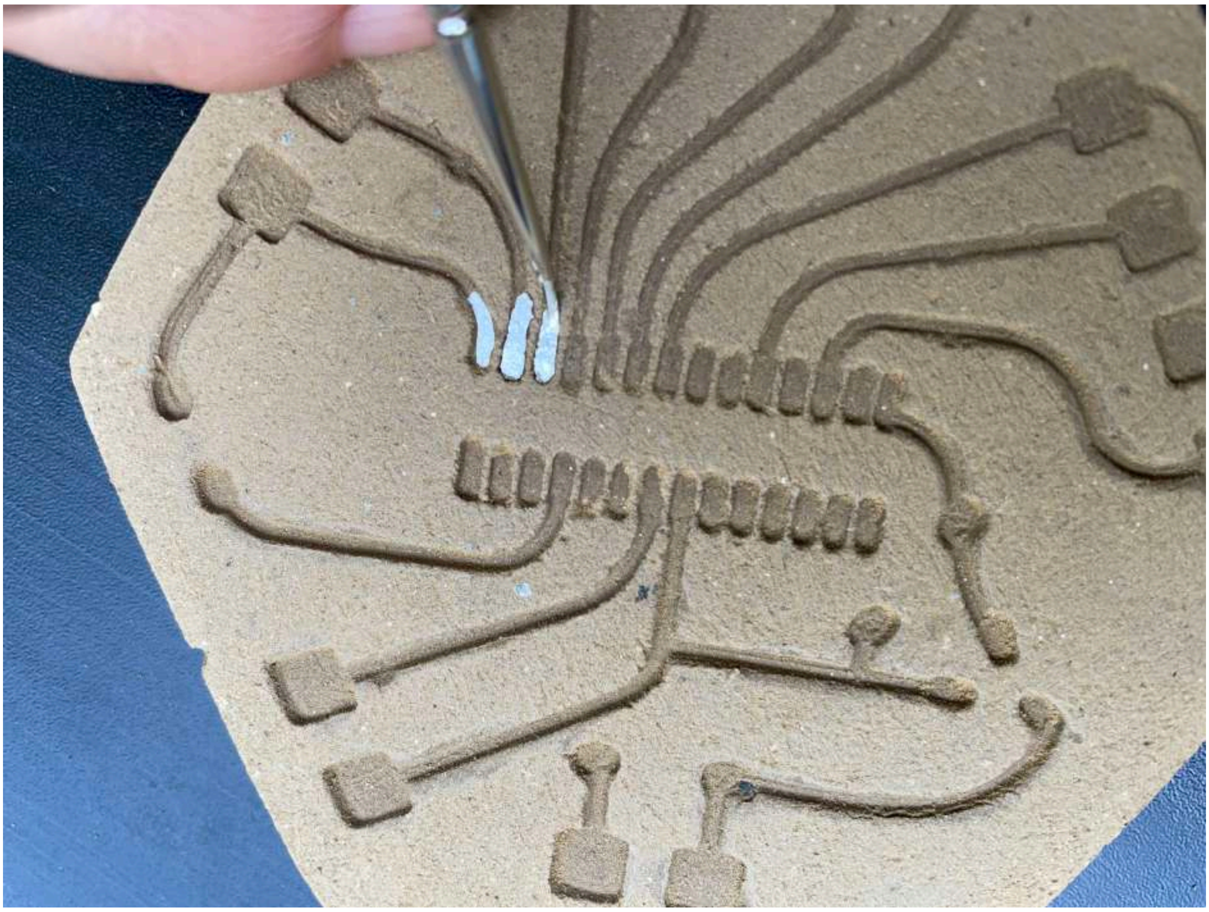
We usually let the boards dry naturally and **outdoors for 24 hours** before painting, but this time-frame is weather dependent. If you have got more time between modeling and firing the clay, you would ideally **dry them indoors for one to two weeks**, positioned between wooden plates and while applying some weight on 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 them quickly you can also place them **around a wood fire**. It is important to **avoid temperature clashes** during drying so the best is to bring them to the fire as slowly as possible step by step.

You will know when a board is 100% dry when you see its color becoming lighter and more homogeneous. When it is not completely dry the edges become lighter color whereas in the middle the clay is still darker and wet. In any case, our experience tells us that this is the minimum drying period required to start to paint the circuits. If you wish, you can facilitate the painting process, by gently **sanding** the boards, using a 1**20 or finer sanding paper**. After sanding, make sure all the dust is out of the board, so you can start with painting.

F - Painting the circuit

While searching for **conductive materials** that can be used in ceramics we came across a **gold lustre** (used often in 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 on its ingredients, especially no information on the sources of the gold and its commodity chain. The second problem was the fact that it is not possible to solder directly on this gold lustre, so we needed to add another precious metal to the equation. The challenge was to find within the **solderable** and **easily available precious metals**, such as tin, copper, brass and silver, one that could bear with the **firing process which is c.a. 700º C** and at the same time keep its **conductive properties**. As we know **tin**, mostly used for soldering, melts at a very low temperature, **copper** melts at approx. 1000C, but the oxidation process happens so quickly in the fire that it loses its conductive properties, the same happens with **brass**. We were left with **silver**, which although it also oxides with the fire, it keeps its conductive properties. Also silver is a cheaper metal than gold and is widely used by goldsmiths. We were able to find a silver paint commercialized by a German company that is made with waste silver powder collected by jewelry makers. It's kind of an urban mining technique of silver dust.

For painting 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. Important in this process is that the lines of silver do not touch each other. If that happens (which always does) you can correct it by using a thin metal piece and scratch it out. For the input and output connector pins, since the paintable area is bigger, we used a thicker brush (0 or 1).
There would have been many other ways to print the circuit avoiding that time consuming hand painting, as for instance using a stencil mask and either spray on it or use another paint transferring technique. The reason we choose this one 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 private backyard, re-using a hole that was previously dug for that purpose. The wood was collected in-situ and it consists basically of dry wood sticks and old branches from our trees. We started a normal fire to gain some heat and placed all the boards around it so they can complete their drying process. In the meanwhile, we collected wood sticks with approx. the same size but in two thickness categories. The ticker sticks can be used for the base of the boards and the thinner to add on top.

While the first fire is lowering, 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 on 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 wood sticks. This will cause the fire to expand and also provide an oven effect for the ceramics pieces. The max temperature should be around 700C but that's obviously hard to control. Our experience tells us that 20 minutes is the average time they need to be ready, so you will need to maintain the fire alive during that amount of time. After these 20 minutes you can let the fire get low by itself and check for the boards. You should be able to see them glow in the fire and that is the exact moment you will know that they are ready. Using the tongs you can now quickly transfer them from the fire to a cold water bucket and leave them there, yet still holding with the tongs for a few seconds. This is usually the ultimate "proof" test for the clay. If there were no air bubbles, stones or cracks, and if it dried properly, it can resist the cold water.

After firing, the silver will turn white due to oxidation, but if you polish it well after, it will turn silver again.

Part 3: Programming (Burn Bootloader the Atmega chip and uploading the program)

A - Burn Bootloader the Atmega328

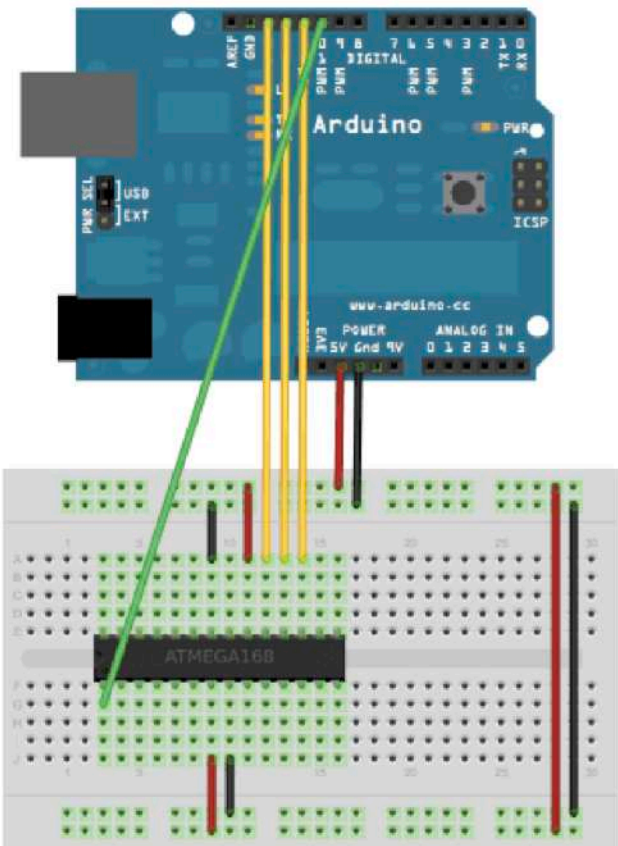
Although the Arduino website has a very good tutorial that explains this step, we will here provide a summary of all the important steps. There are two ways depending on two different situations: Either you just want to migrate and upcycle your Atmega from an old Arduino UNO board or, you have got a single loose Atmega chip (without an attached Arduino board). Our tutorial requires the later, using a minimal circuit and eliminating the External Clock of the Arduino (avoiding adding a 16 MHz crystal to the circuit) and using the Atmega328 internal 8 MHz RC oscillator as a clock source instead. Beside the Atmega328 you will need an Arduino board, a breadboard and some jumper wires.

A1 - 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 just copy here the instructions from the page.

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 its necessary location in the Arduino preferences dialog). If you've previously installed support for additional hardware configuration, you may already have a "hardware" folder in your sketchbook. Otherwise 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 in order to burn the bootloader onto an ATmega on a breadboard using the 8MHz RC internal oscillator, follow the diagram:



A3 - Burning the Bootloader

To burn the Bootloader follow the steps copied and edited instructions from the Arduino page after wiring everything as the diagram before.

1. **Upload the ArduinoISP** sketch onto your Arduino board (you find it on Arduino > File > Examples > 11 Arduino ISP. You'll need to select the board and serial port from the Tools menu that correspond to your board.)
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**

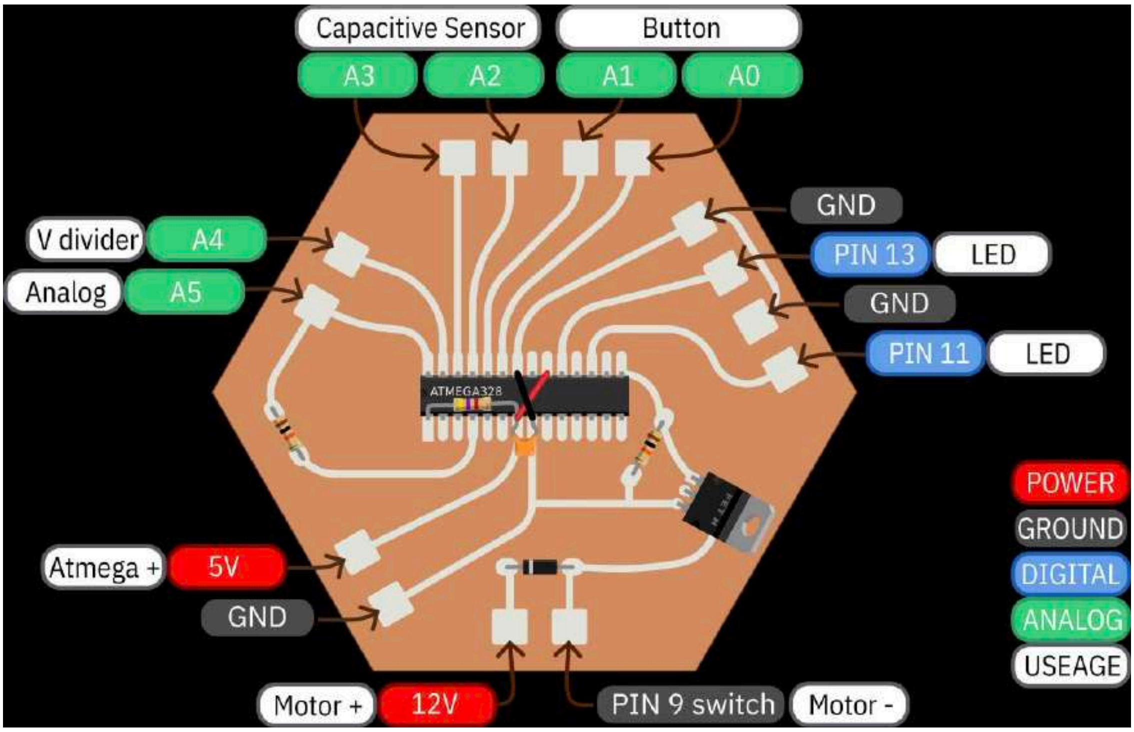
Note: in case you encounter errors please follow this threads for troubleshooting:

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

B - Uploading the program

B 1 - Understanding the program:

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

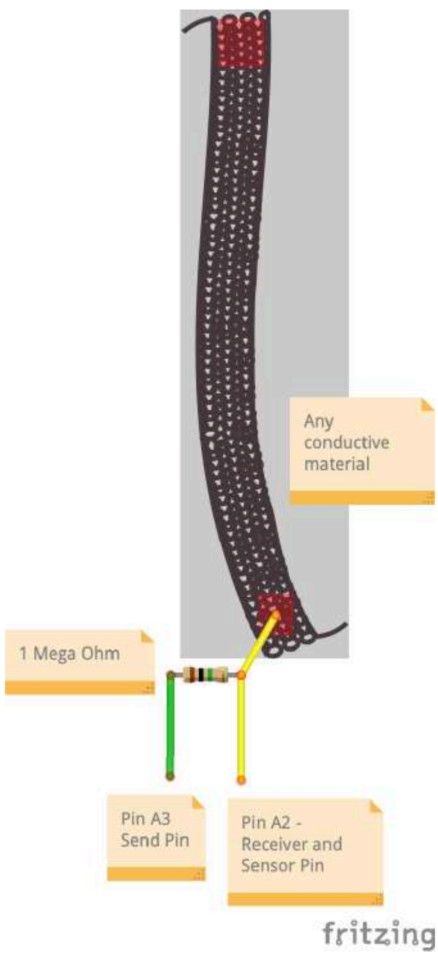


The digital **input "Button" button (Pin A1 + A0)** is ON whenever the circuit between the 2 inputs is closed. Clicking on the button will immediately affect the Digital outputs (**Pin13 and Pin11**). To these pins we can attach an **LED or a Piezo buzzer** (a DIY speaker could also work but would require more electronics to amplify the signal). This means that if I **click on the button** repeatedly in 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 affected "distorted" by the **Capacitive Sensor (Pin A3 + Pin A2).**

For programming 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 the sensor Pin A2.**

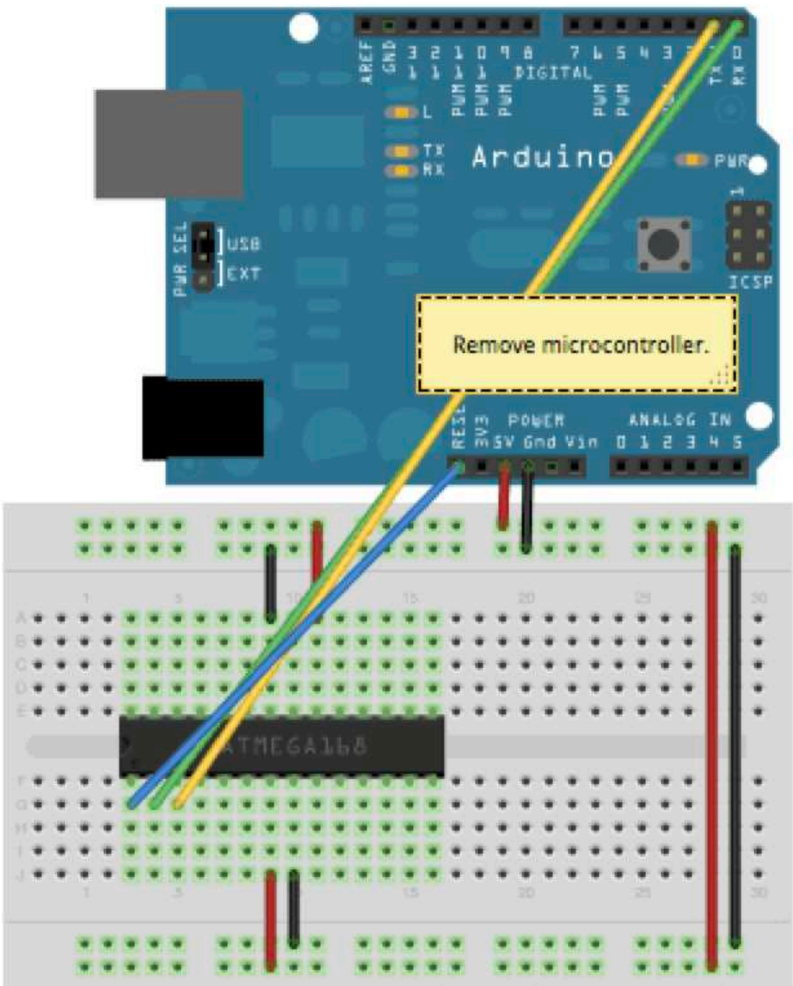
You can use any conductive material to do a capacitive sensor (any metal, copper tape, e-textile).



When clicking the button for more than two seconds, it automatically switches to the motor output. A **12 Volt DC motor** can be directly connected to the **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 **analog A5 Pin** and **voltage divider A4** will affect the **speed of the motor**. Note: you can use any analog variable resistive sensor (a normal Light Sensor LDR or even a piece of fruit) directly connected to Pin A4 and A5, but take into consideration that you might have to replace the resistor on board for better results depending on the resistance of your sensor.

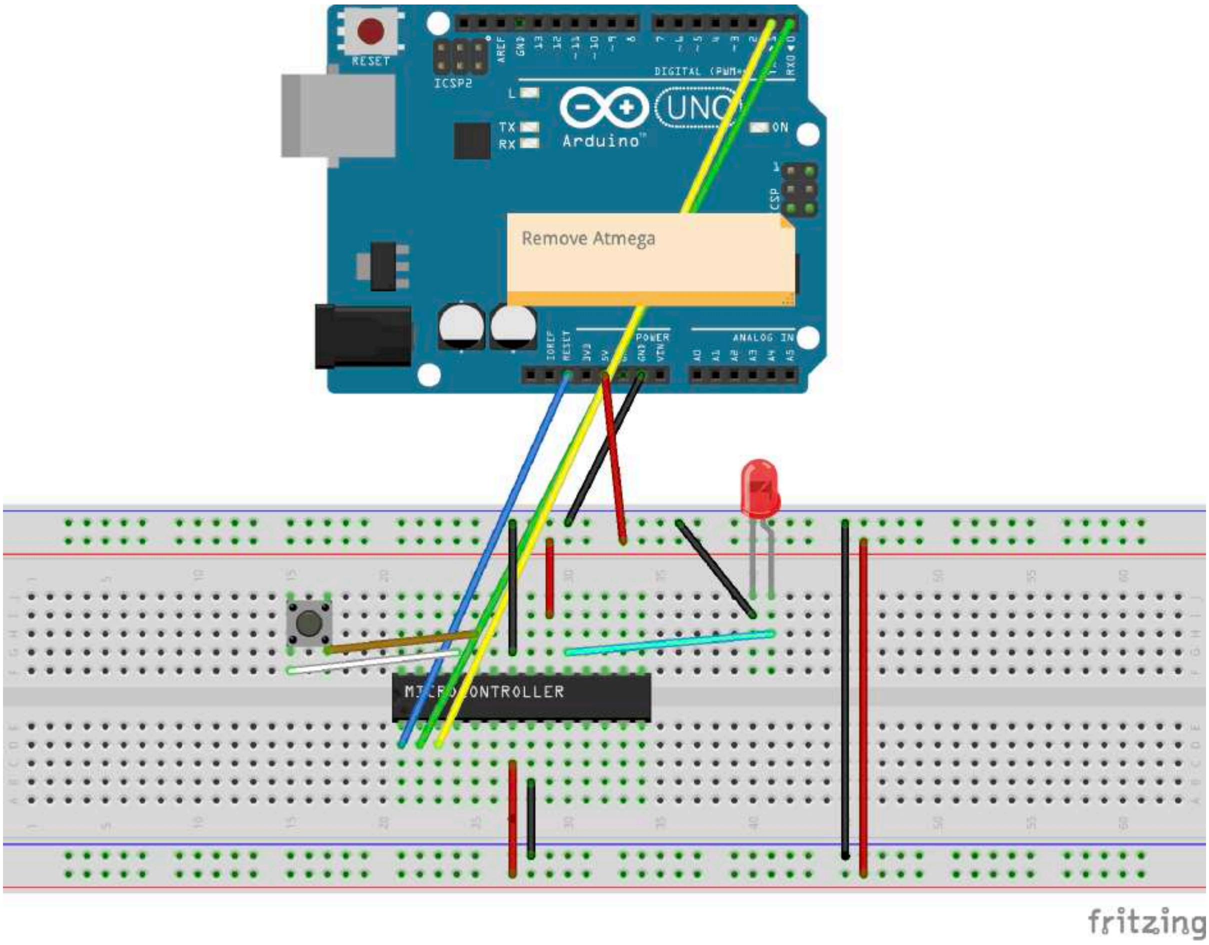
B2 - Uploading the sketch to the ATmega

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 just 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 of the Arduino Board. Open the sketch in the Arduino IDE and before uploading make sure you select from the Tools > Board menu**."ATmega328 on a breadboard (8 MHz internal clock)"**



B3 - Quick test

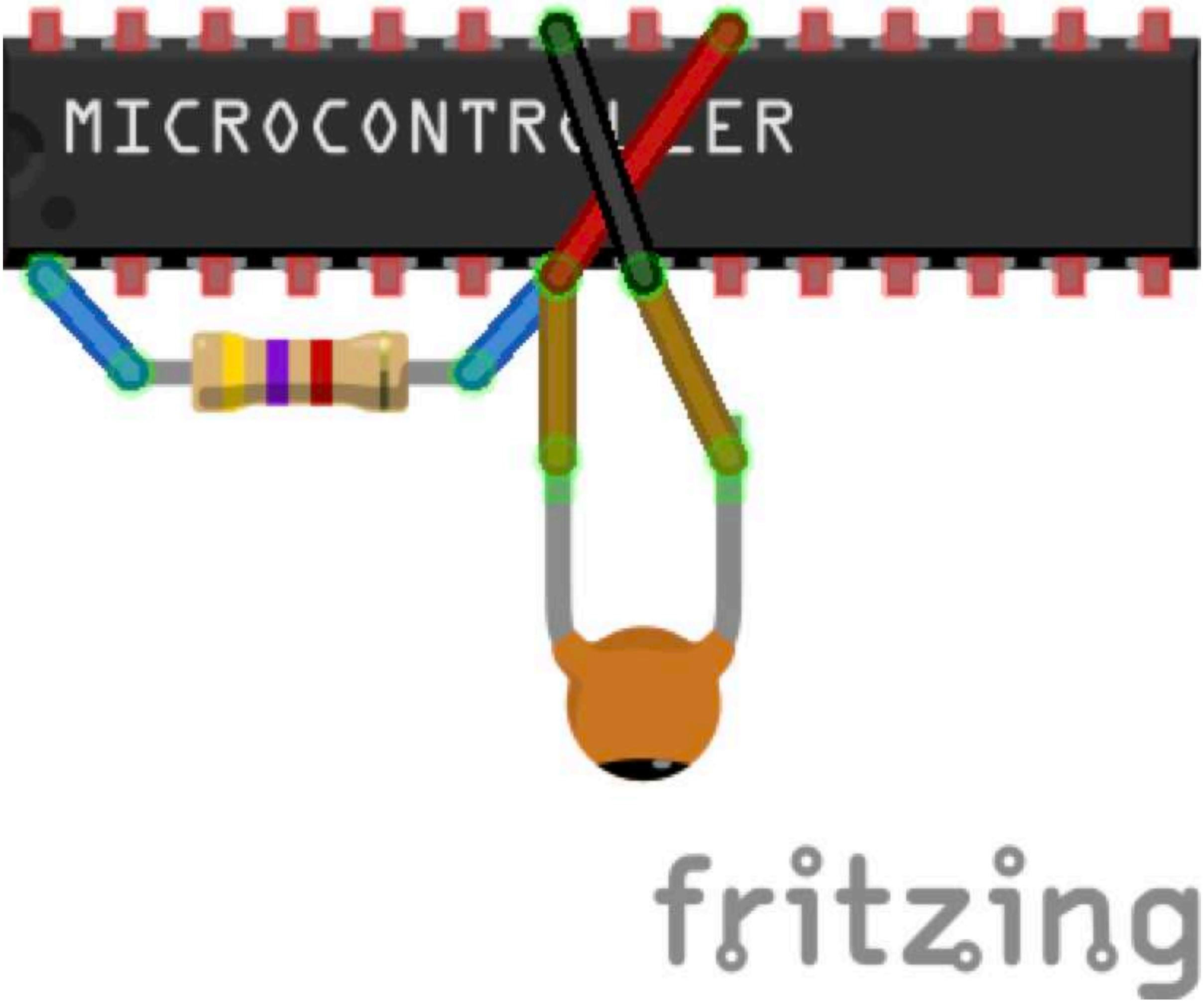
You can quickly test if the program is working by connecting the **button A1 and A2** and an **LED** (Light Emitting Diode) either to **Pin 13 or Pin 11**. If the LED flickering reacts to the rhythm of clicking the button your program should work.



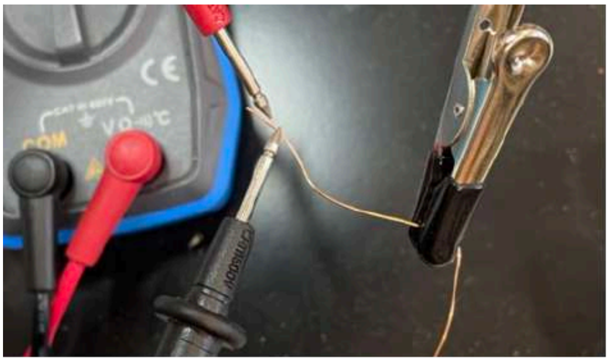
Part 4: Electronics (Soldering the electronic components)

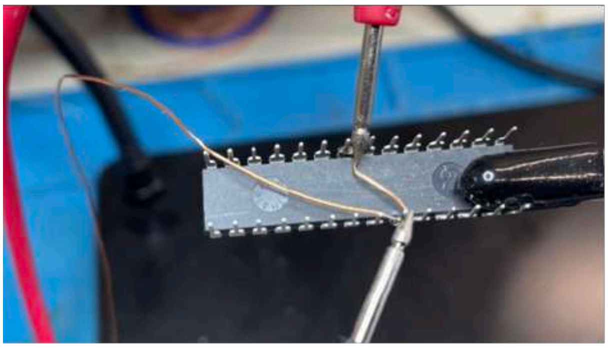
A- Preparing the Atmega

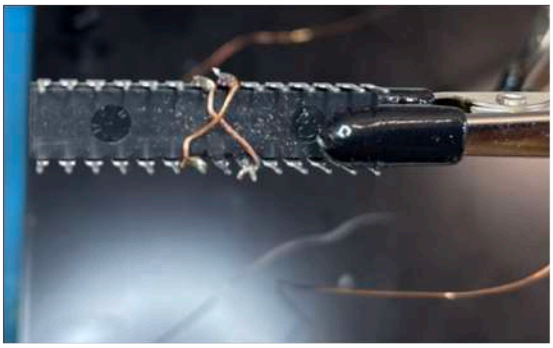
Before soldering the components to the board we prepare some of the connections directly on the Atmega. We need to **bridge across the chip the VVC Pin and Ground**, and additionally add a **4.7 K resistor between the RESET Pin and VVC**, and a **100nf ceramic Capacitor between VVC and ground**. (We opted for this step to avoid crosses in our single layer printed circuit.)



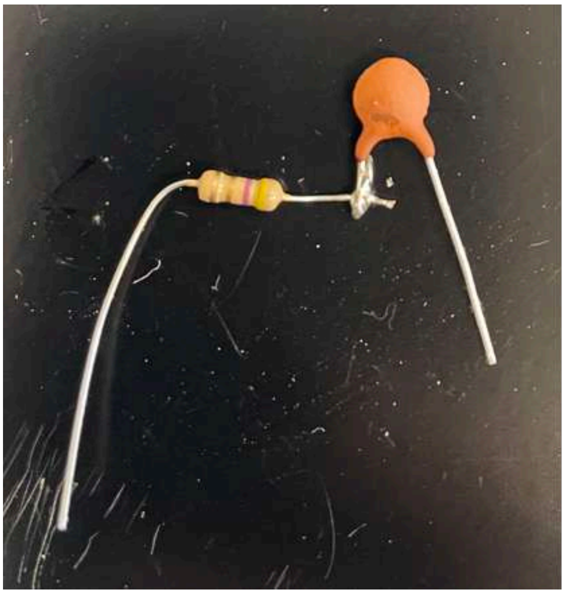
To avoid many wires on top of the Atmega we used a very thin coated copper cable (in this case we had one in our workshop) to connect **VVC to VVC** and **GRD to GRD**. The wire we used is insulated to make sure the **wires cannot touch each other** causing a short circuit. Before soldering it to the right pin, it is necessary to sand out a little of this coating material in the edges you desire electrical contact. Before soldering it to the chip, **test for conductivity** by using a multimeter and repeat the process after soldering.

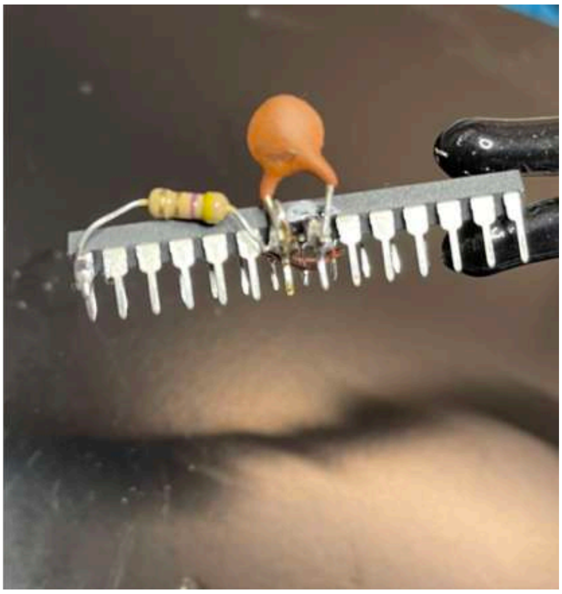






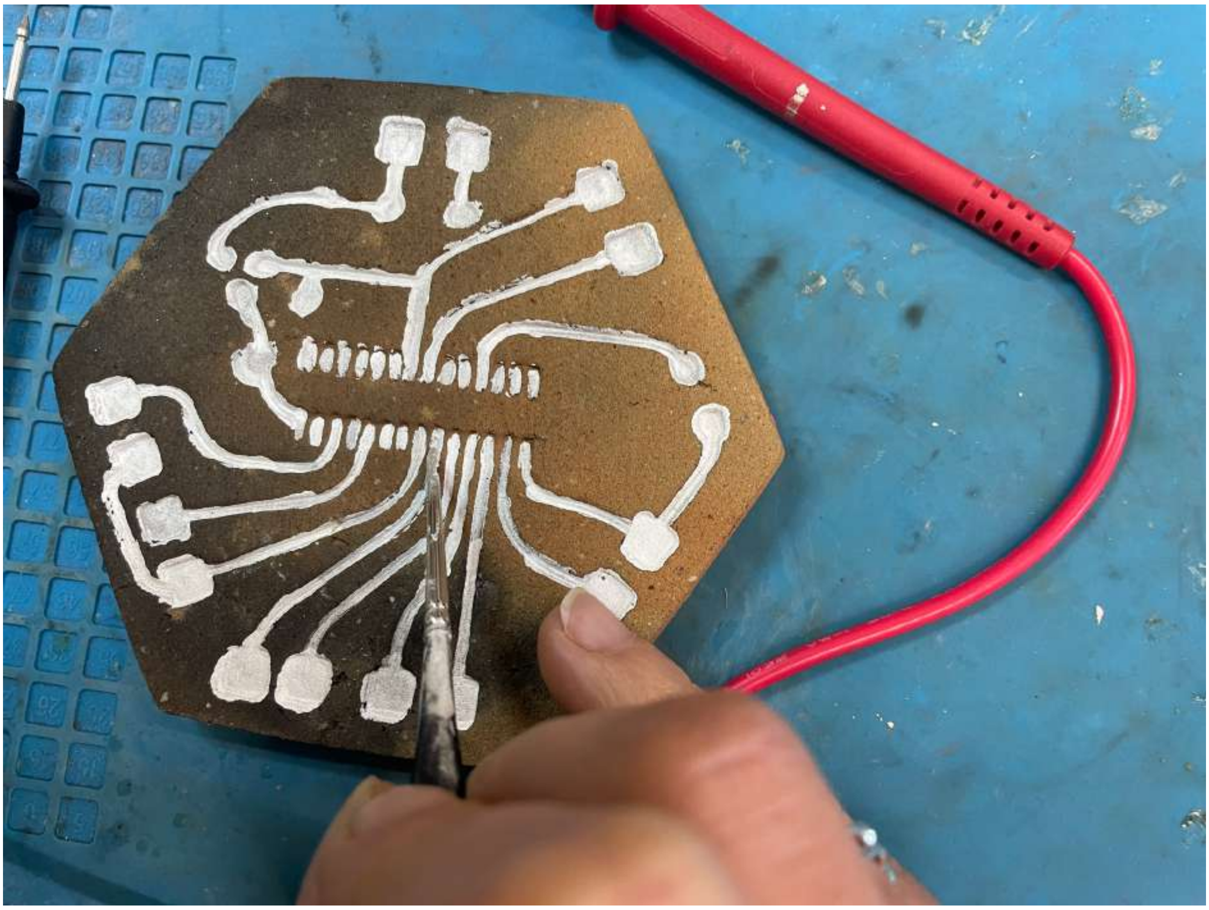
We joined the **capacitor with the resistor** before adding them to the Atmega.





B- Adding components to the board:

B 1. In the first we **check conductivity** by using a normal **multimeter**. In case you find small parts that are non conductive you can add a little silver paint on top to correct it.



B 2. All our **parts are recycled from old electronics.** Here a list of what you will need:

- **Resistor** 10 K (x2)
- **Schottky diode**
- **Power Mosfet N-Channel** to power the 12V motor (We use the IRLB 8721, but a similar component can be used as long it has got the same wiring diagram - see bellow)

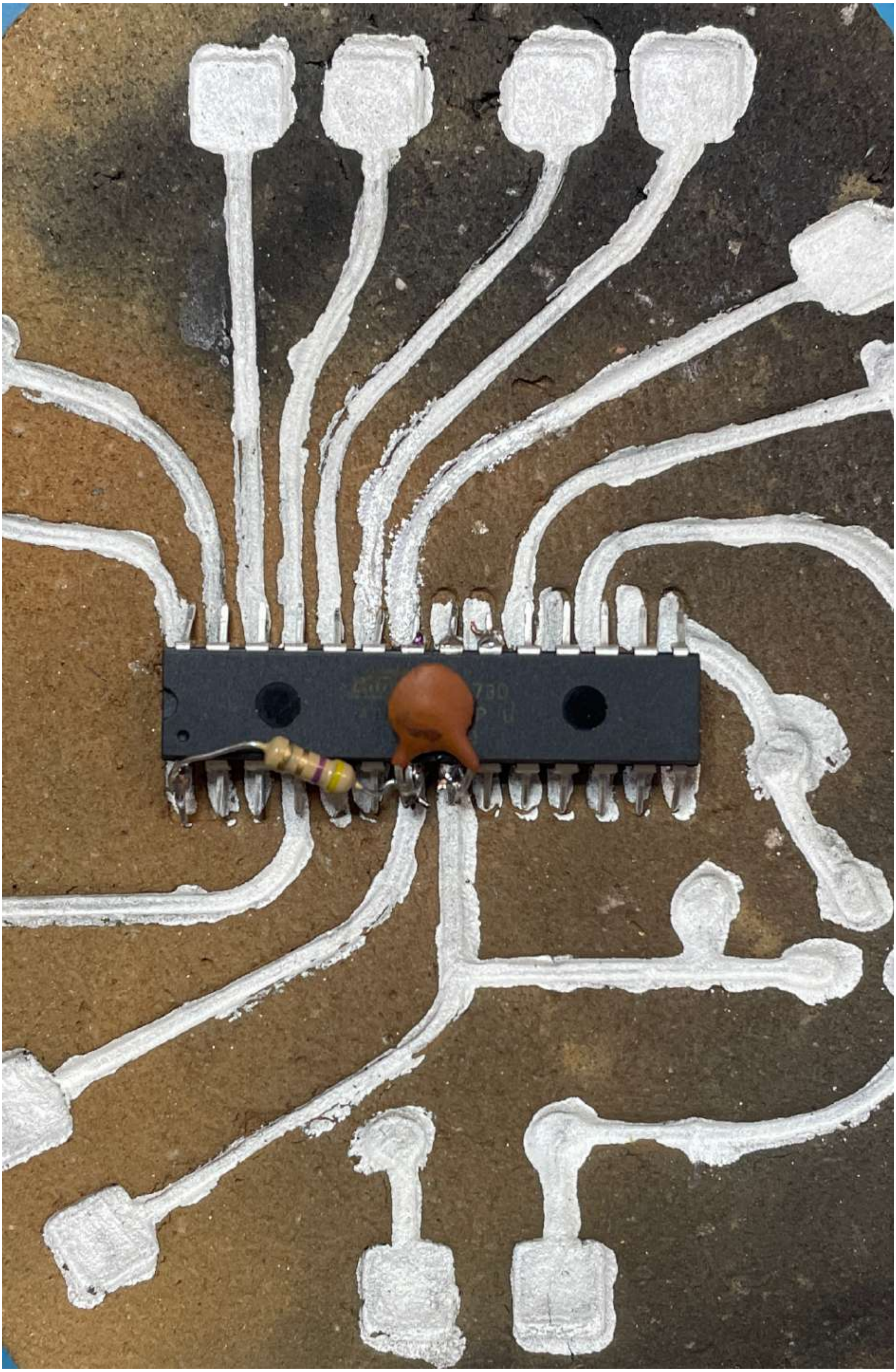
G	D	S
Gate	Drain	Source

B 3. Soldering

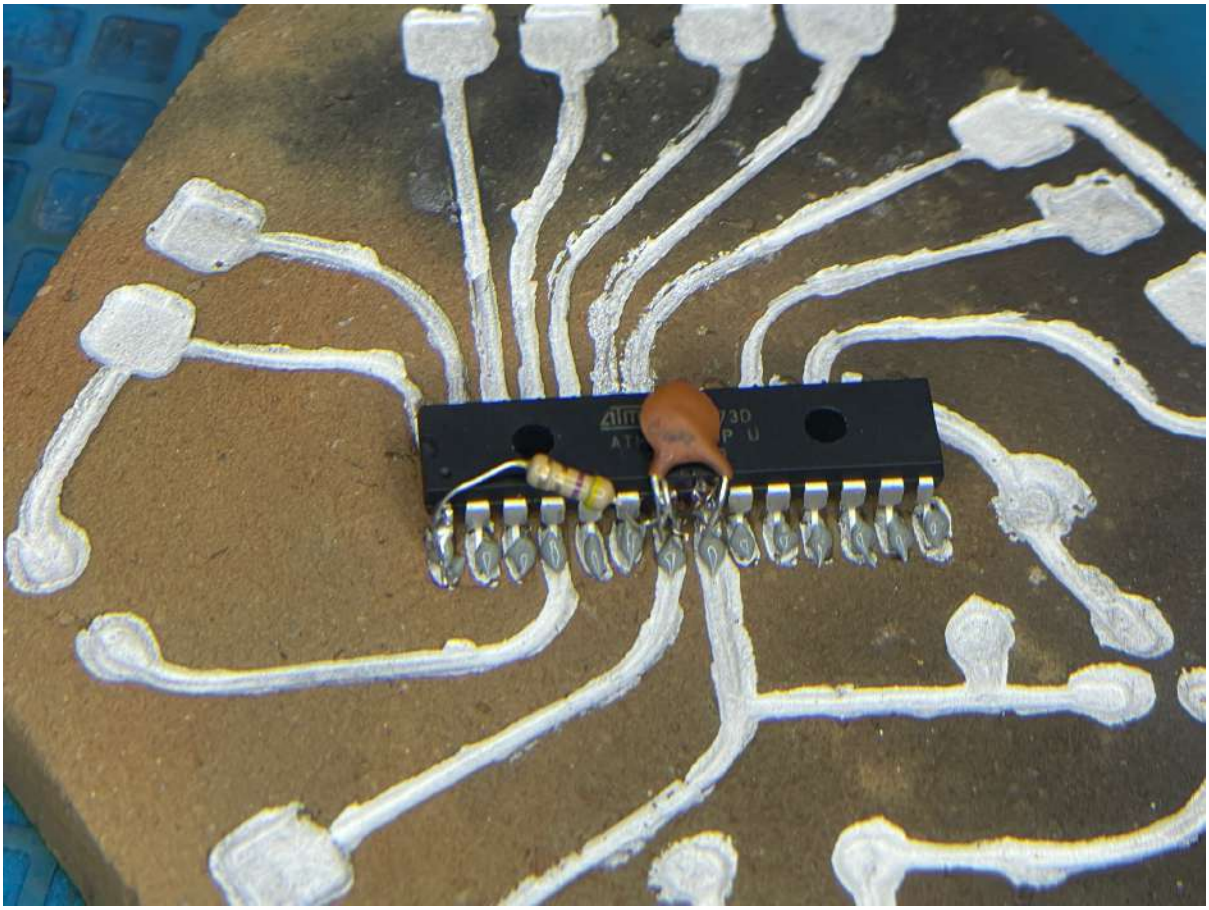
Because the silver connectors on the ceramic PCB are very sensitive we concluded that the **SMD soldering** method with **hot air soldering** would be more appropriate. We used a **lead-free SMD soldering paste** ISO-Cream ® "Clear' from Felder, a Germany brand that produces under a fair-trade standards,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

Before soldering make sure 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 **start with the Atmega** by bending the connectors in a way that fits 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** on each of the Atmega pins.

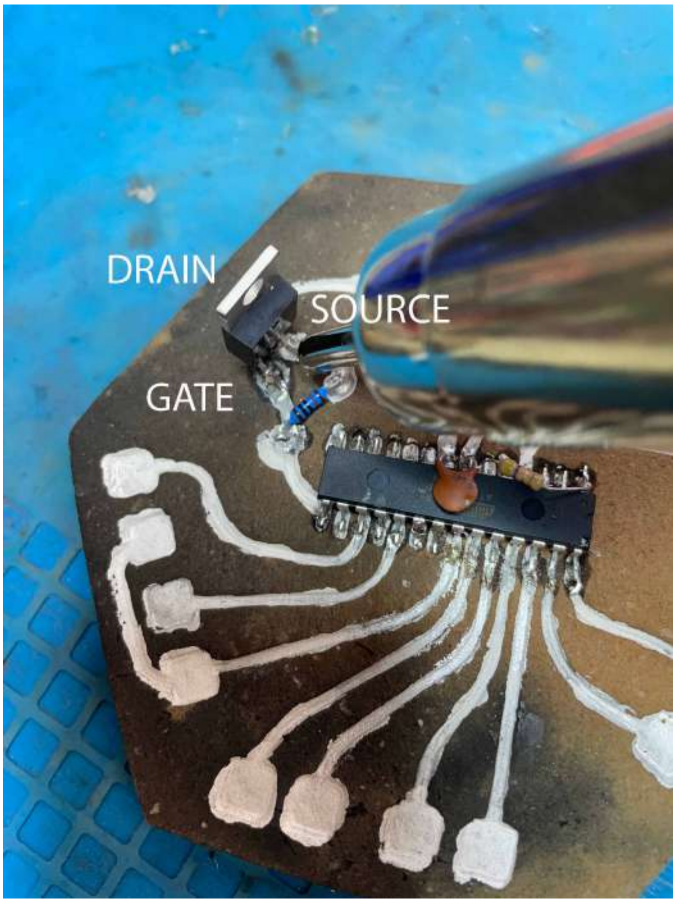


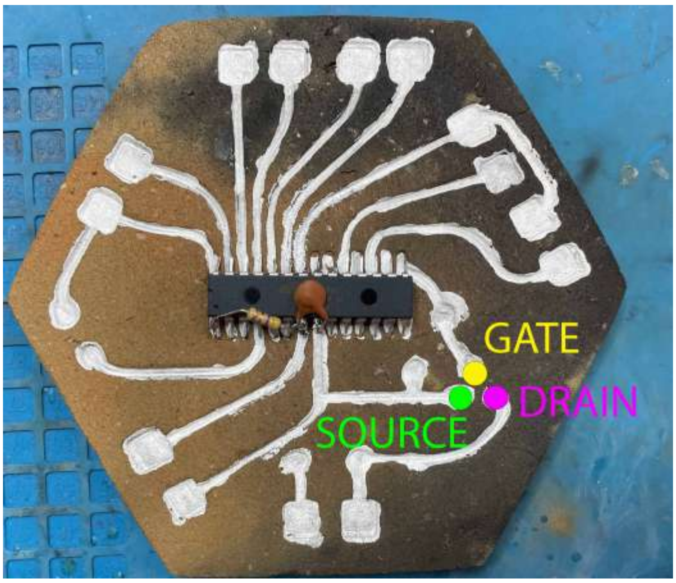
Make sure that the solder paste is not touching two connectors. After verifying all pins you can start to hot solder it.
Pay special attention when using hot air soldering; since we don't see the heat we tend to cross our hands and tools over it. Be in particular careful in this process and do everything slowly and patiently. When you're ready just approach 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





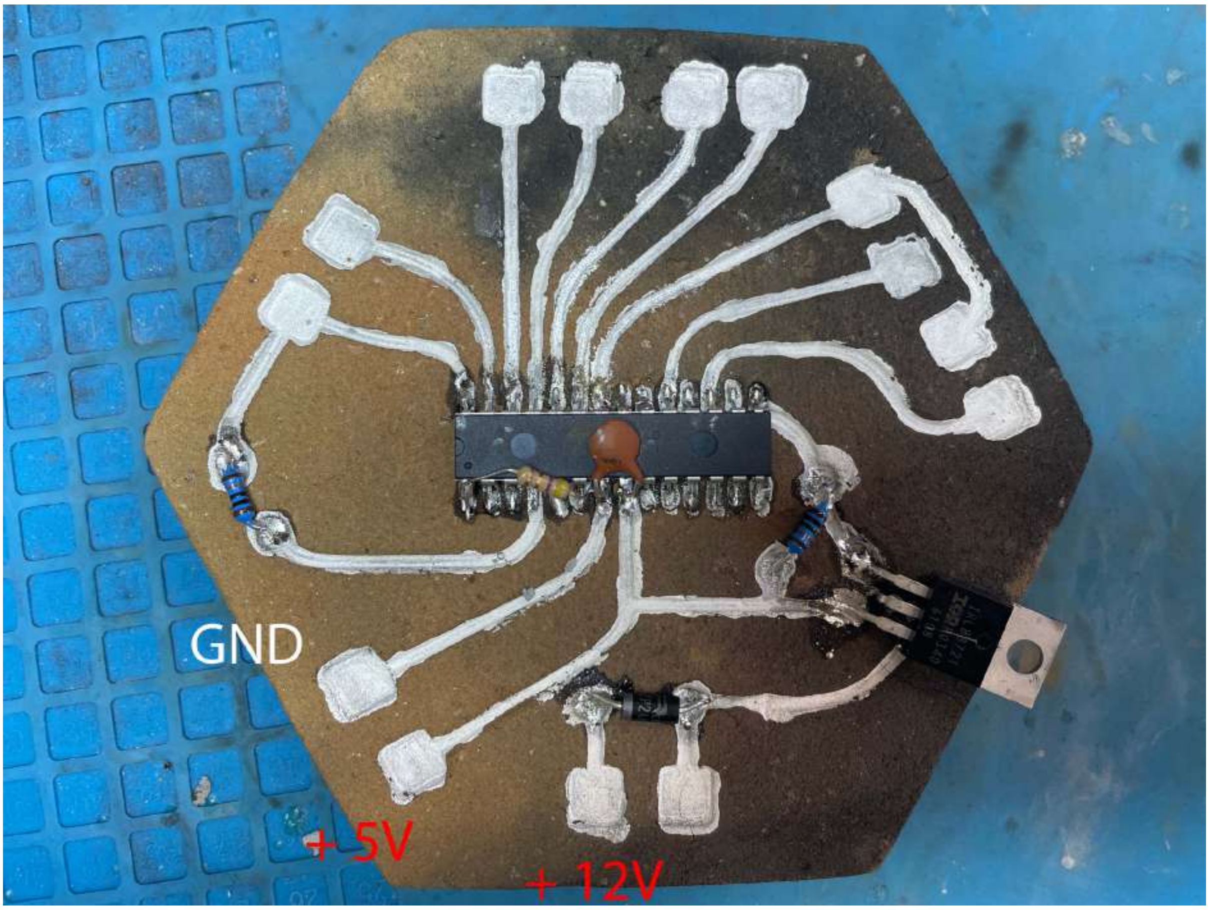
Part 5: Powering and testing

A - For testing the board your will need two power connections:

POWER 1 Atmega - 5V DC

POWER 2 Motor - 12V DV

Follow the picture for connecting correctly the power:



In order 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 come up with another solution. For this problem we could not find a fair-trade or ethical 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 to leave it to dry overnight.

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