Functional Reflowing Hot Plate

by ana_carolinaa

The aim of our project is to develop a hot plate for reflow soldering printed circuits boards with Surface mount Technology (SMT). There is a curve of Temperature versus Time, that help on the control that how many degrees is going to warm and how much time it is necessary for each material.

The temperature curve associated with the reflow is an essential parameter to control the correct connection of parts. The parameters of certain components will also directly impact the temperature curve. Therefore, we choose the Ramp Profile Chart, it is easy to control the temperature and the process has really good repeatability.

In this attached file, it is possible to check all the instructions to use the machine and the warinigs about it.

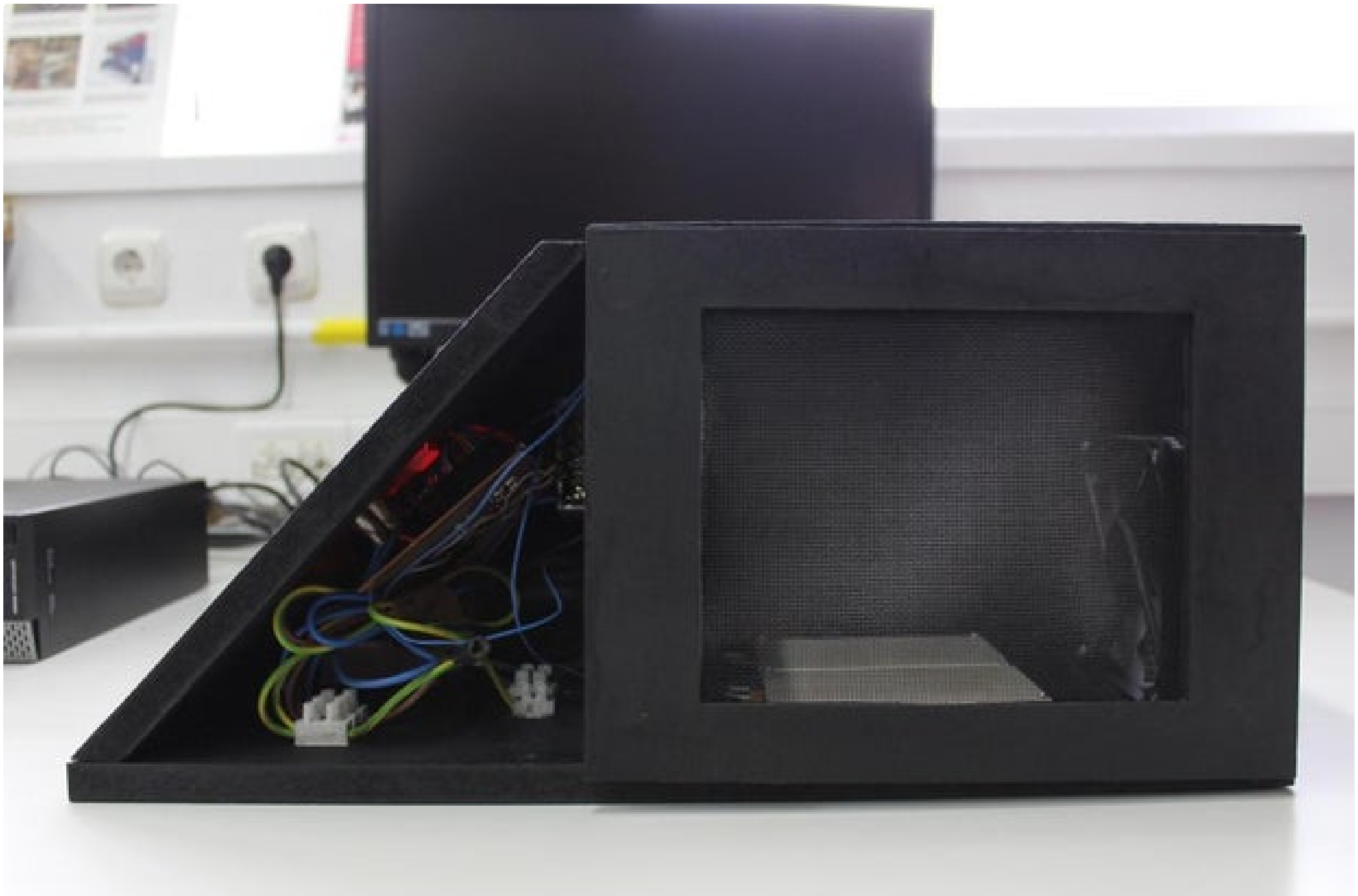
Supplies:

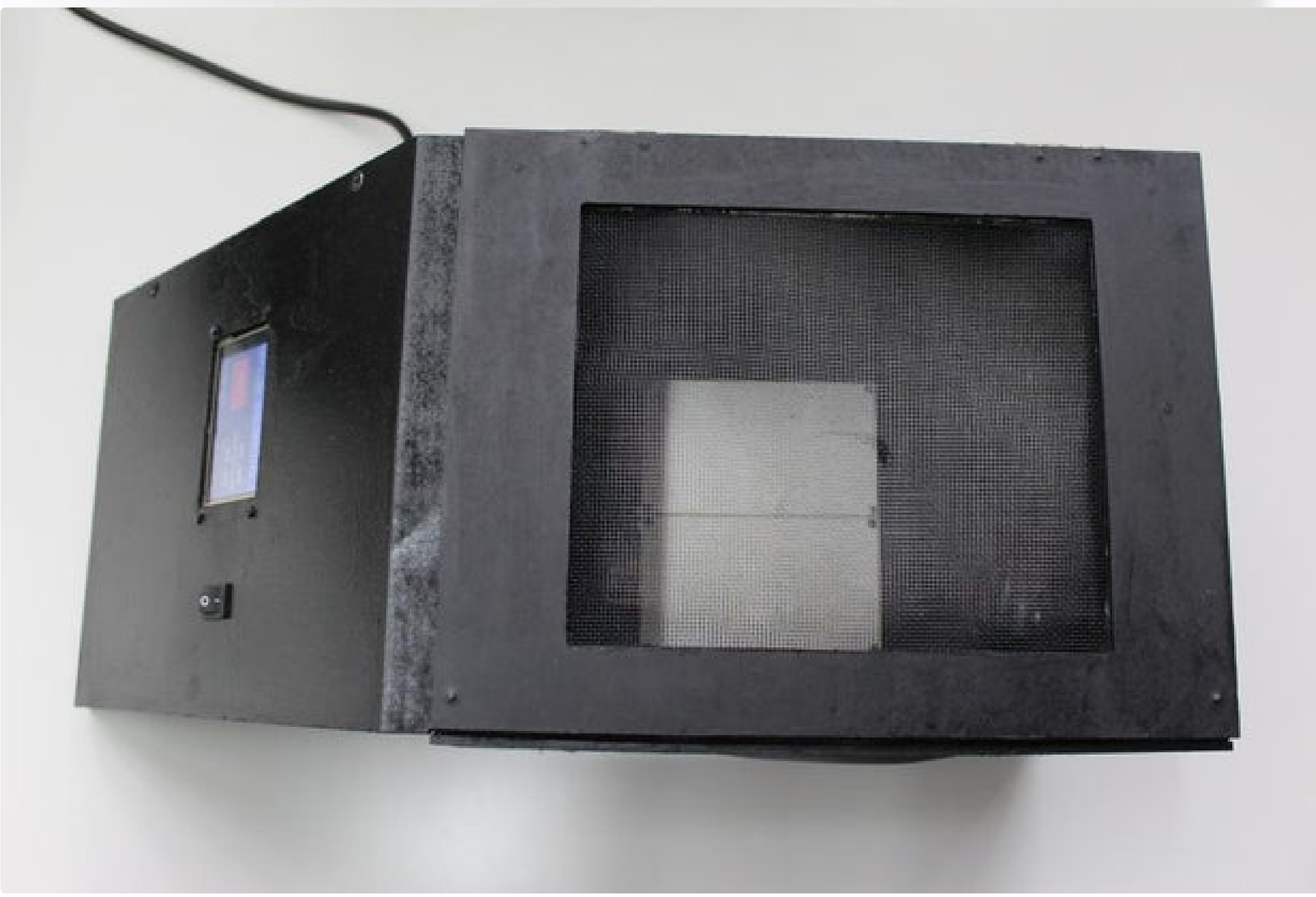
Materials

This is the list of all the materials that we bought for our project:

- Converter AC/DC 5v
- Sintetic wool Aisolaytor
- Relay SSR 220v-5v
- Termopar Sensor of Temperature 0~600℃
- AC Port
- Arduino (nano)
- Hot Plate 400W
- Display board
- Buttons to choose each mode
- Aluminum Sheet
- Interruptors
- Isolater Tape
- Thermistor NTC 10KOHM 3977K BEAD 300° C
- High temperature resistant silicone
- Conglomerate (600x300mm)

In this attached file it is possible to see some links of most of the materials.





Download

https://www.instructables.com/FA0/HCI5/LIHFOCYV/FA0HCI5LIHFOCYV.pdf

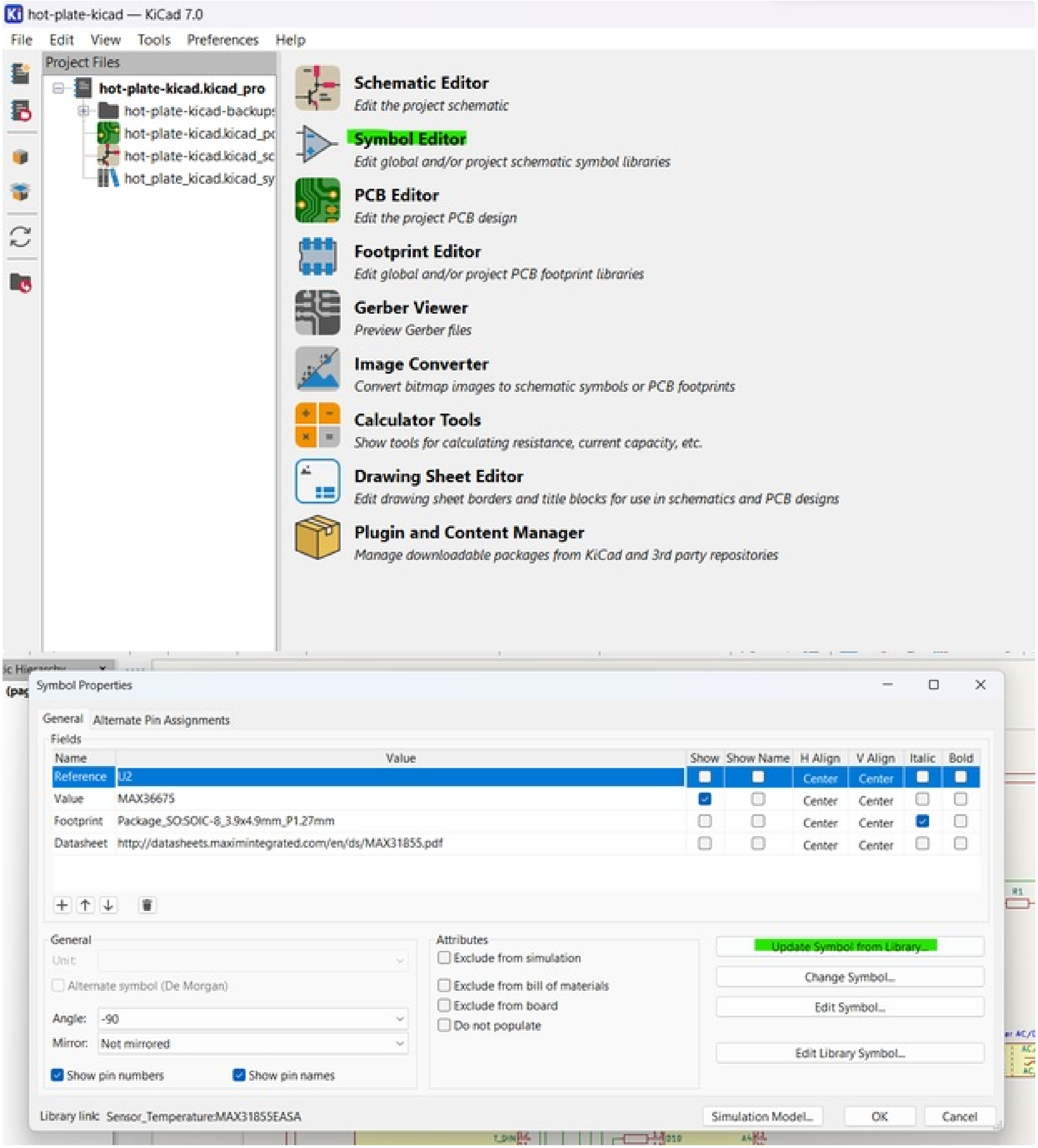
Step 1: Learning How to Use KiCad for Your Project

In this step, we started our electric circuit on a professional platform called KiCad. This plataform allows you to make a good schematic for your project, and this helps in assembling your project.

First, in the *Schematic Editor* we looked for the most similar components on the general library that contains in KiCad. (marked in green on the second image). In this library it is possible to find every kind of component for your project, has a good diversity.

After that, if it is necessary to add a new component, at Symbol Editor in the main page of the plataform you can create your own project library and add all the components that are missing. All the components are saved in your library, and whenever necessary you can add a new one according to what you need for the project. (third image)

Another function that is really important on KiCad, is that when you change something on your library, you can go to your component on the schematic and just Update Symbol from Library, this allows you to change your component without having to redo it in the scheme and harming some other part. (if you click on your component, in the right corner shows the button to update - third image)



Step 2: Case and Mechanical Part

We start cutting to the measurements that have been planned in the scheme with the help of the miter (table saw). After each cut, the surface is sanded to obtain a clean result since the material breaks down easily when handled roughly.

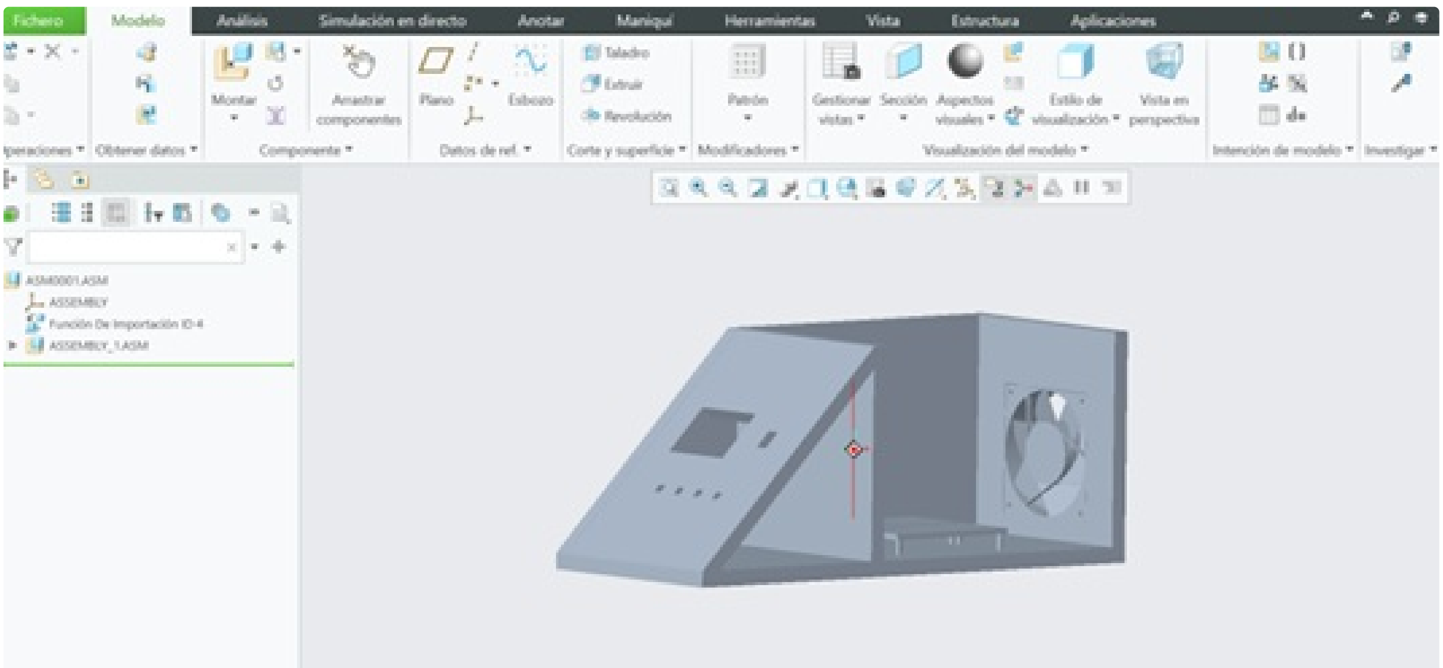
First of all, we work the part of the control panel. We proceed to lower the central area to 3 mm to be able to enter the power button, the screen and the accessories it requires. Apart from that, it will be necessary to lower at all the zones that will contain the components.

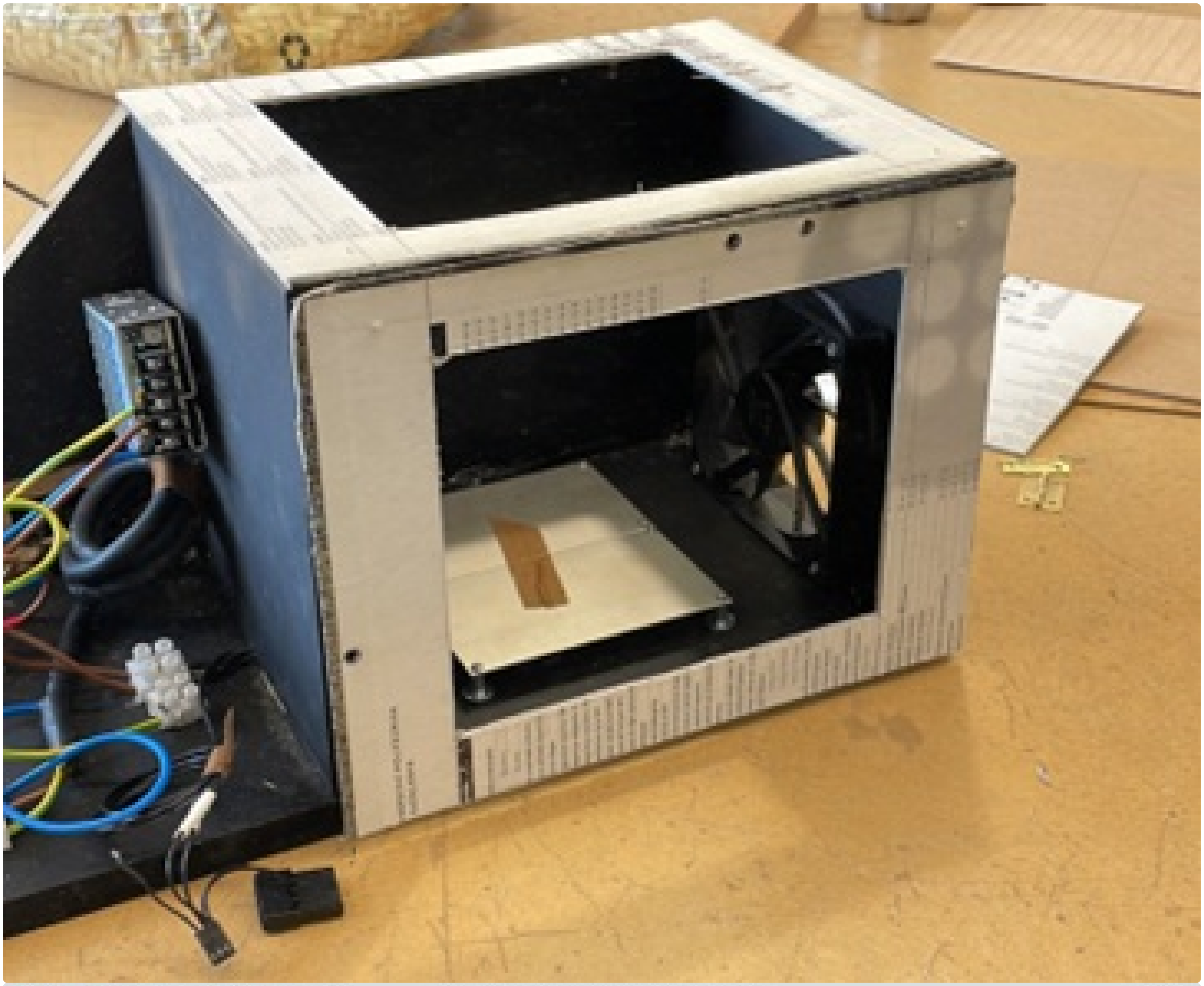
Next, we work on the 200x200mm surface where we must place the fan. We must open a through hole which allows the air flow to enter and touch the hot plates as much as possible. We opted for a cut shape with rounded edges to obtain a better fixation of the fan; it can be seen in the following figure. (Image 4)

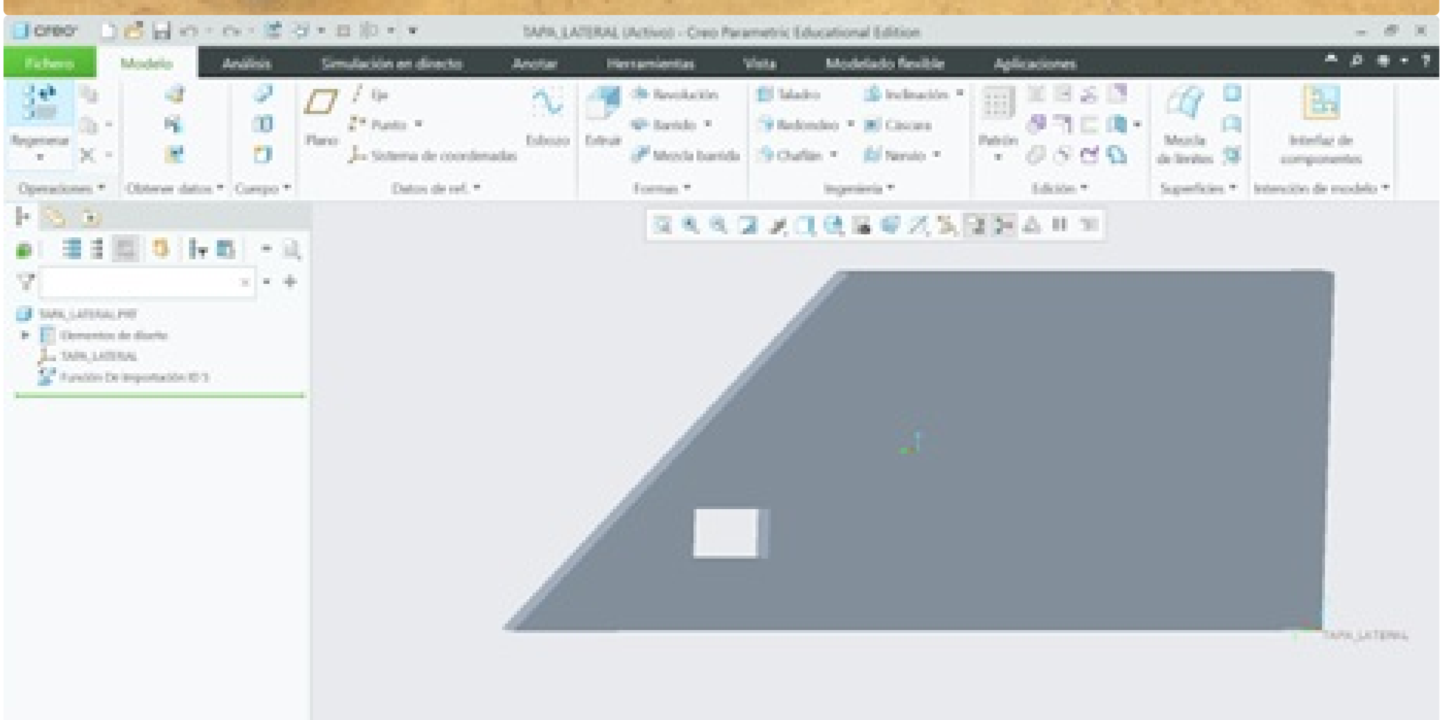
Next we proceed to open a through hole in the area shown in figure 3. The purpose of this will be to hold the plug of the system.we use tools such as the drill, the keyhole saw and the polisher.

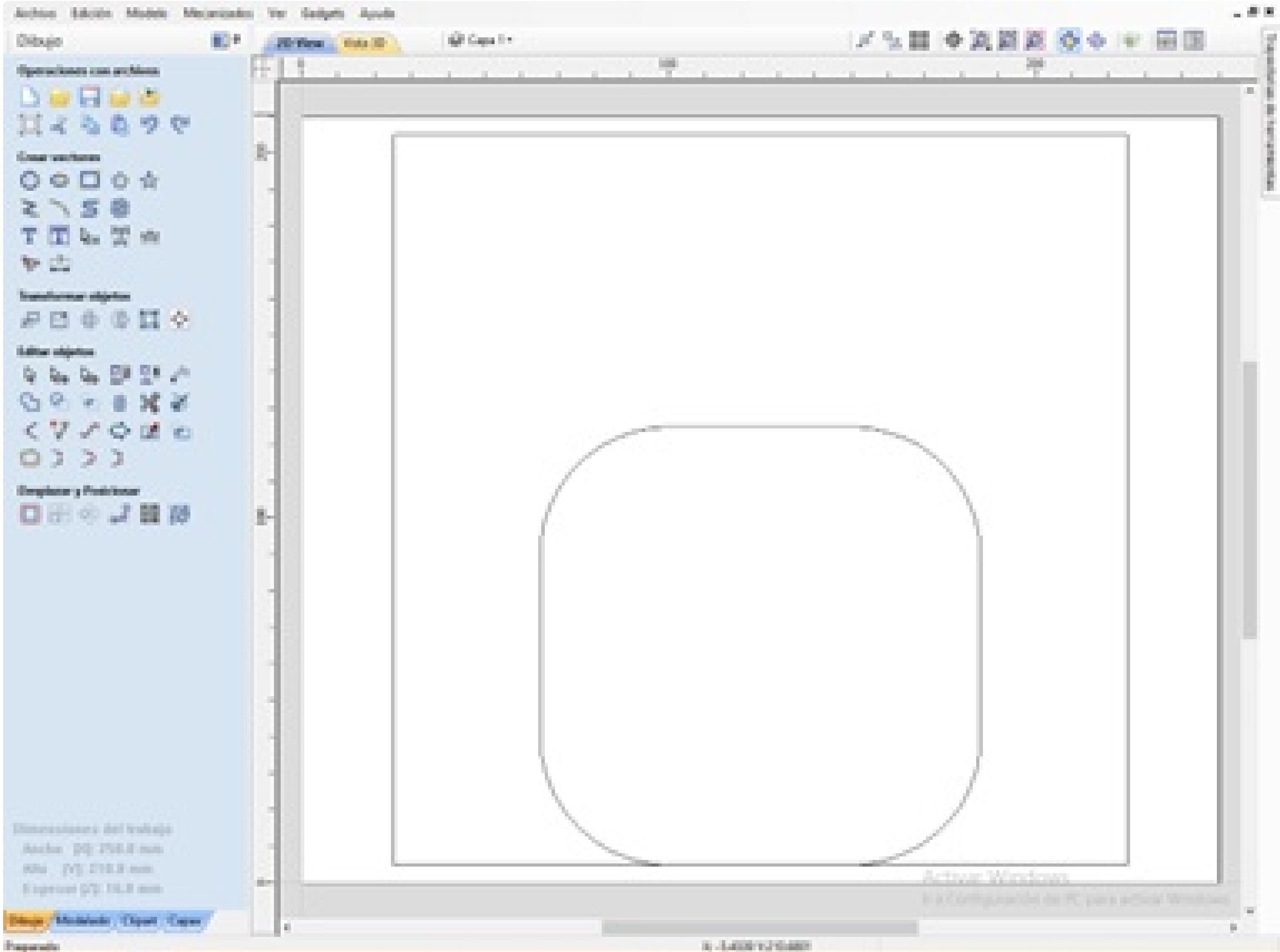
We opted for the previous system to save time in manufacturing, since as we have verified the milling process consists of many long and durable parts. Another reason why milling was not used in this case, was that it wasn't predicted that we would need anchor zones to fix it and they were not done.

The safety cover is made of conglomerate wood on the upper part and hard cardboard on the lower part, leaving a hole in the central part that allows the air flow to drain inside correctly. (Image)






TAPA_LATERAL (Activo) - Creo Parametric Educational Edition
Fichero
Modelo
Análisis
Simulación en directo
Anotar
Herramientas
Vista
Modelado flexible
Aplicaciones



View in 3D Download

https://www.instructables.com/FEF/19PK/LIHFOCSJ/FEF19PKLIHFOCSJ.stl

Step 3: Electronics Part

For this step, we had this basic sketch of our circuit just for helping in the assembly. The Arduino gets the temperature from a thermocouple, whose signal is amplified by the MAX6675. It then displays the temperature on an LCD and switches a Solid State Relay (SSR) if needed. It is all connected acordding to the schematic on KiCad.

First, we connected the arduino to the amplifier called MAX6675, which is connected to a termocouple that inside has a alloy material that when is connected to the hot plate generates a voltage, because of this material is also possible to amplify the voltage and make hotter the hot plate.

About the low voltage:

Since they don't draw a lot of power, we can connect everything to the Arduino pins and configuring the required pins for power and ground. The Arduino is powered via the USB port, so we connect it via a USB cable to the power brick.

About the high voltage:

In this part, you have to be very careful when connecting all the stuff to the hot plate, make sure everything is desconnected when you starts to working in it.

To connect the arduino to the amplifier, we used 3 pins: D2, D3, D4 (of arduino) and CS, SO, SCK (of amplifier), the conexion is like this:

D2 ---- SCK

D3 ---- SO

D4 ---- CS

The amplifier is also connected to the pin 5V(for giving power) and GND of the arduino. (Check the schematic on the image 4)

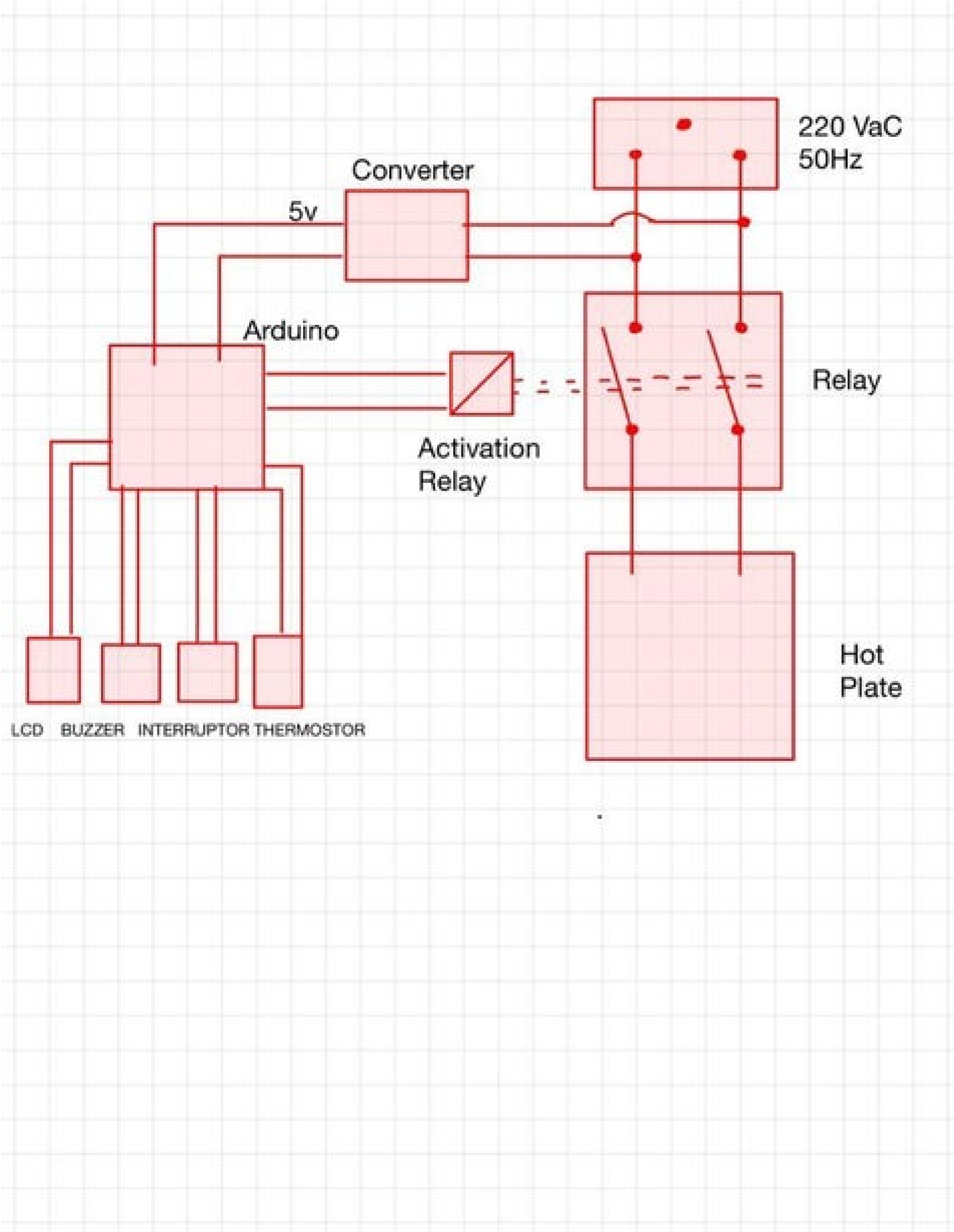
In the second part, we start connecting the relay to the IEC port, the hot plate, the converter AC/DC of 5V and also the interruptor turn on and turn off.

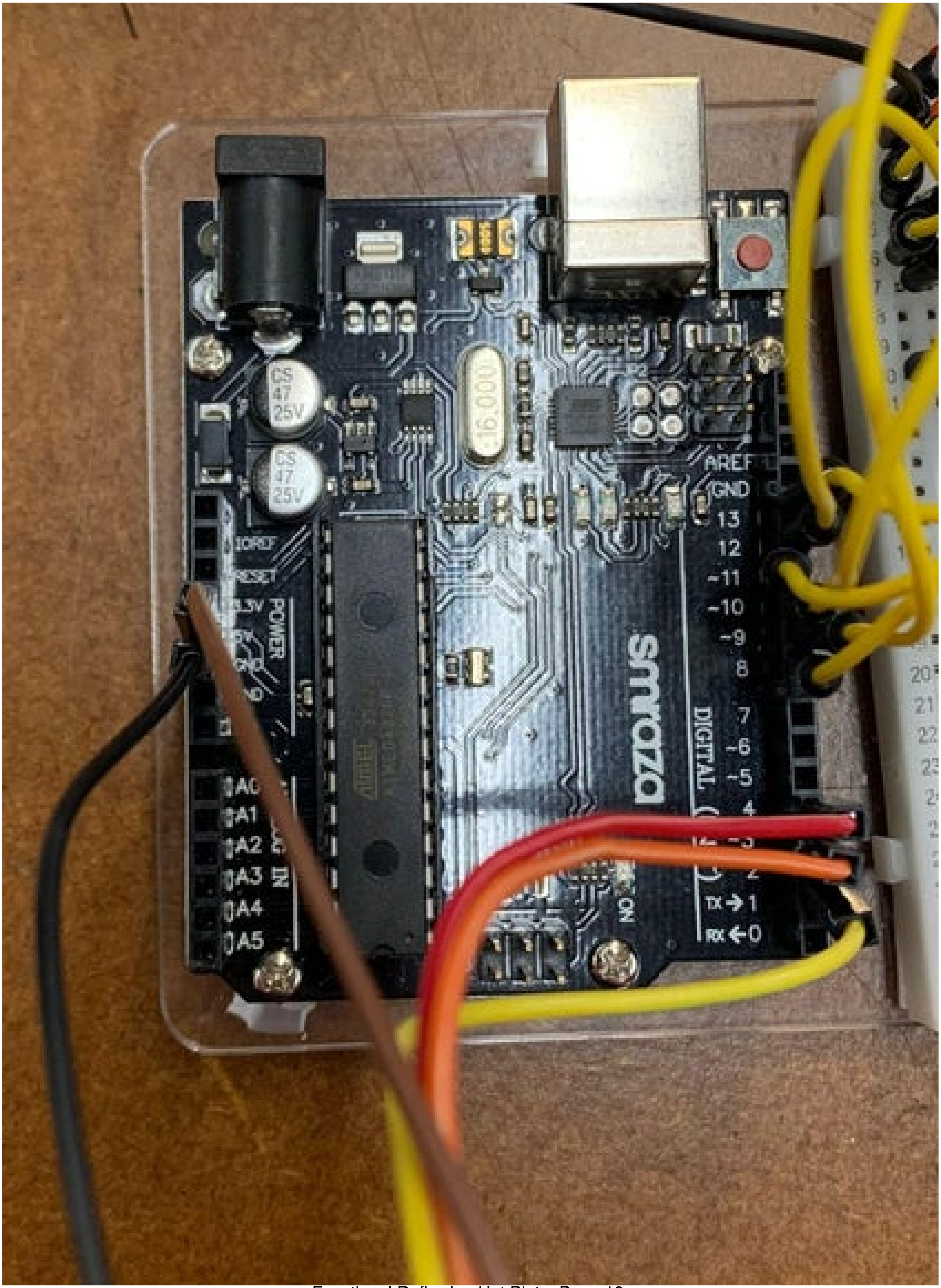
In the IEC Port, there is *a line, an earth and a neutral.* We connected the *line* to the pin 1 of the relay and also to the pin one of the interruptor, after that the pin 2 of the interruptor was connected to pin 2 of converter AC/DC. The *neutral* one, we connected to the hot plate and also to pin 2 of the relay. And the hot plate connected to the pin 2 of the converter AC/DC. This all look like this:

Besides that, in the converter AC/DC we connected the pin 3 to a GND, which is a point of reference for tensions in a circuit. The pin 4 of the relay, it is connected to a 5v power, with that it "opens" and "closes" to get hot the plate.

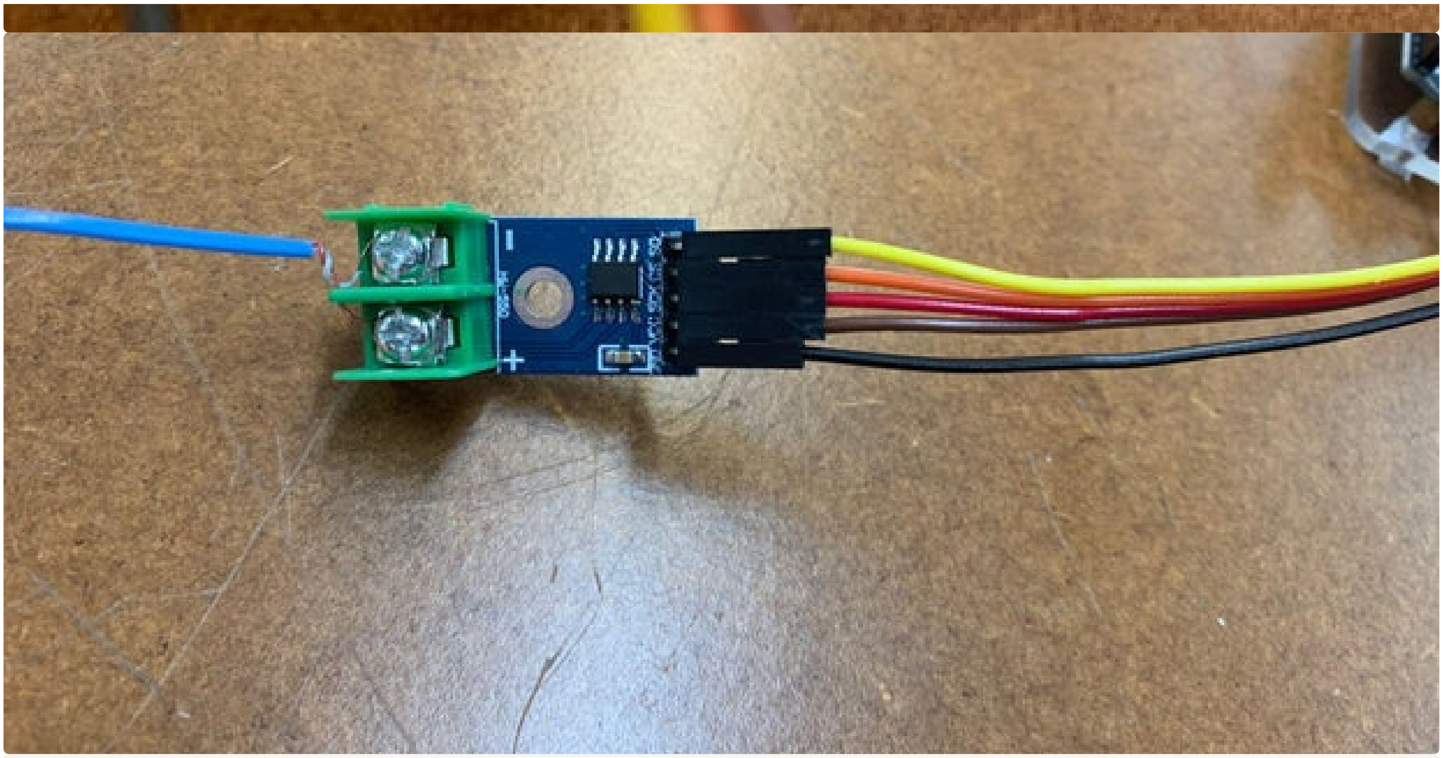
In this reflowing machine there is a fan, when the target point its is slower than the real temperature, the fan starts working to help on the cooling process.

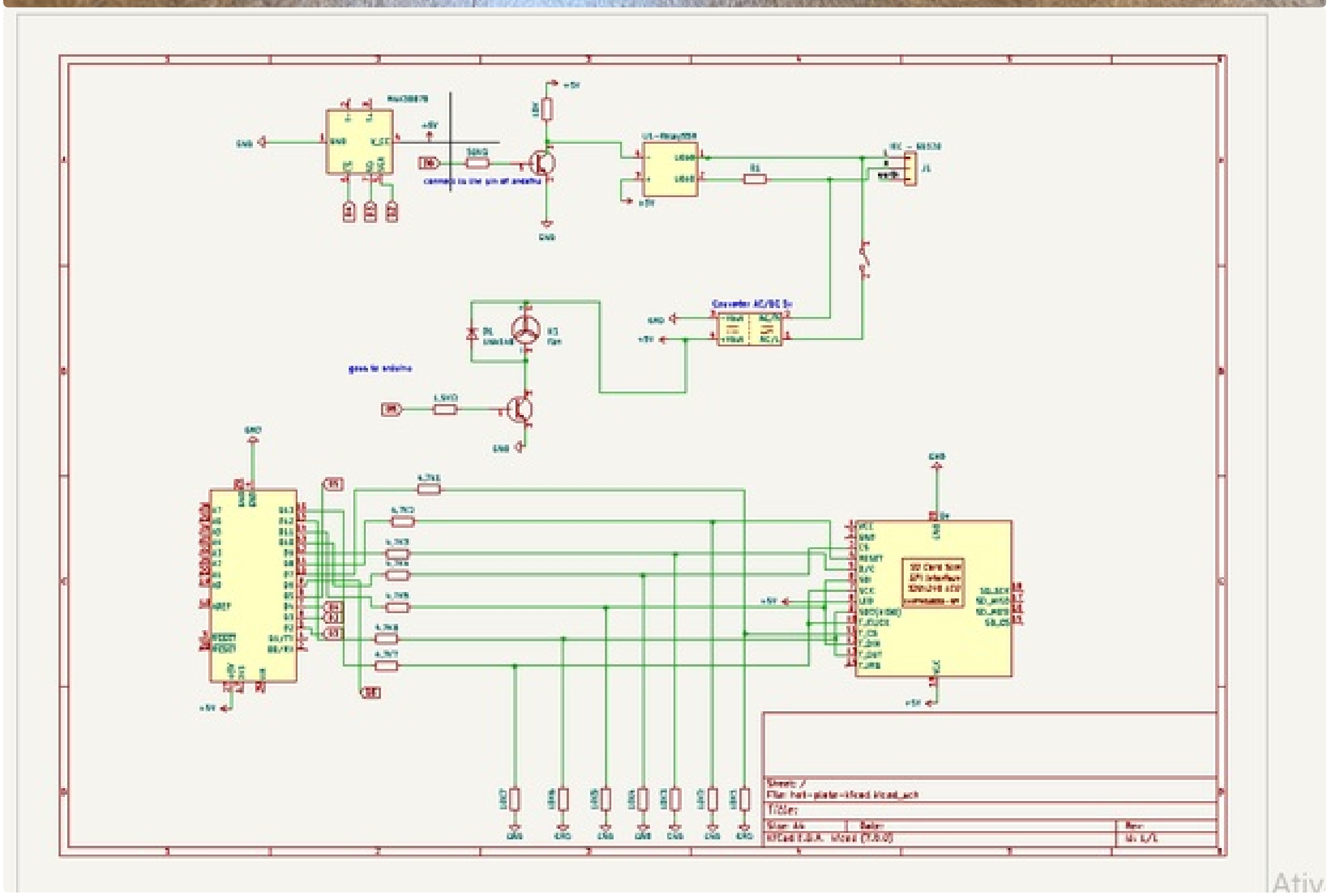
It is possible to connect all of this following the shematic, that allows you to see all correct


220 VaC
50Hz
Converter
5v
Arduino
Relay
Activation
Relay
Hot
Plate
LCD
BUZZER
INTERRUPTOR
THERMOSTOR



AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL
IOREF
RESET
3.3V
POWER
GND
A0
A1
A2
A3
A4
A5
ON
CS
47
25V
CS
47
25V
16.000





Step 4: Programming

To do the programming of this relowing hot plate the temperature needs to follow a specific profile, that is called **reflow profile.** The code allows to store multiple profiles to satisfy different needs, prefered lead-free solder. There is some caracteristics about the profile that should be considered:

- The preheating zone should increase the temperature at a *maximum* rate of 3 °C/s. The purpose of preheating is to allow the solvents to evaporate and to activate the flux.

- The soak zone brings the temperature of all components and board areas to an equal level. Due to differences in thermal inertia, components do not heat up at the same speed —this is especially the case with infra-red heating, due to uneven absorption of infra-red energy by components (depending on colour and reflectivity).

- In the reflow zone, the temperature rises at a rate of approximately 2 °C to a temperature above the melting point (the *temperature above liquidus*, or TAL). The peak temperature for the reflow zone is at least 25 °C above TAL, because solder both coalesces better with the copper and *wets* the pads and component pins better when it is hotter —thereby creating better joints. This is especially important for lead-free solders, because this solder is less effective at wetting than leaded solders.

- The flux in the solder paste gets activated at approximately 100 °C. After it is activated, the TAL should be reached within roughly 3 minutes — or less for no-clean solder paste, as as these pastes have less active flux. Exceeding that time deteriorates the effectiveness of the flux. In other words, the flux in the solder paste sets an upper bound on the duration of the soak zone.

With that information, we created some values for the reflow profile

mpProfile[] = {21, 150, 183, 217, 240, 240, 217, 183, 150, 100, 60};

int timeProfile[] = {1, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300};

double currentTime = 1.0;

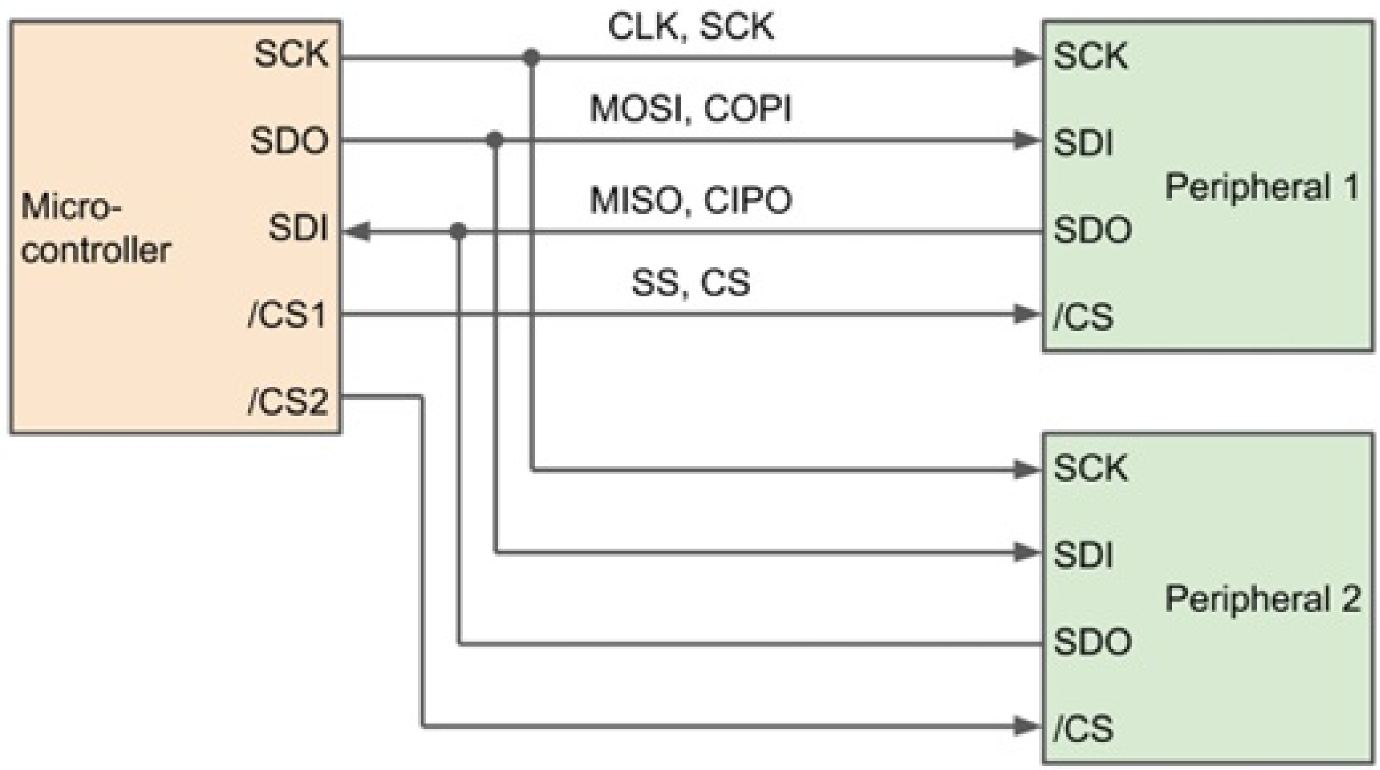
The *mpProfile* is how much temperature will get hot, and the *timeProfile* it is in what time will be each temperature. So it starts on 21° in 1 second, 150° in 30 seconds and follows like this. With this defined it is possible to follow the graphics and how the hot plate heats up.

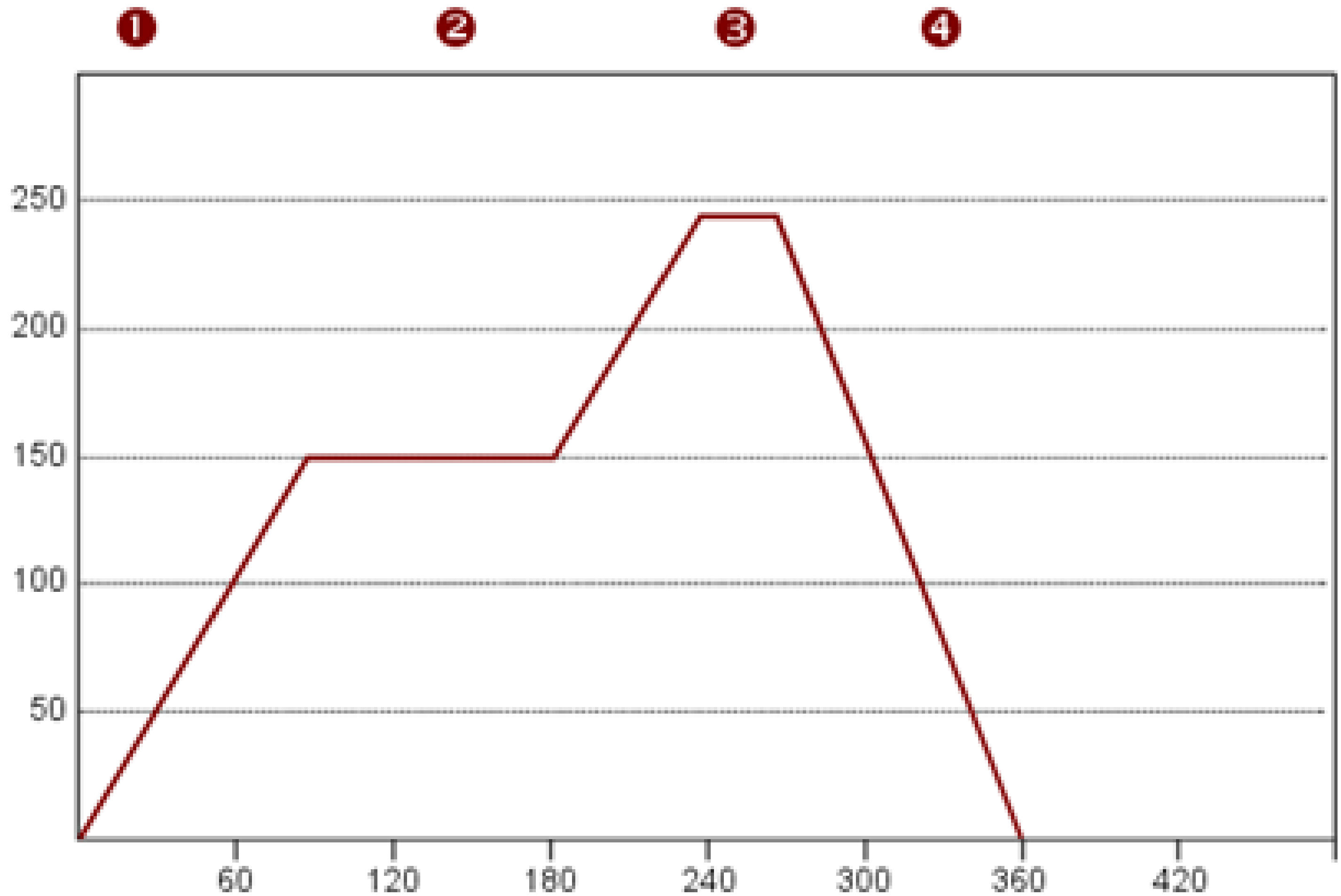
The temperature profile has been created setting some specific points in two lists, one for the temperatures and the other indicating the time. With this information it has been really simple to create a temperature profile as this one. (Image 2)

To the **controlling part**,the heart of the control system is a microcontroller, specifically the arduino nano. It serves as the central processing unit, responsible for executing the control algorithms, monitoring sensors, and coordinating the operation of the different components. The microcontroller receives input signals from temperature sensors, such as thermocouples, which measure the temperature at critical points within the reflow oven.

Furthermore, the control system may incorporate user interfaces, such as a touchscreen, allowing operators to monitor process parameters. The user interface is designed to be intuitive, facilitating ease of operation and enhancing the overall user experience. It has a nice frontend design with a button that allows the user to start and stop the soldering operation and shows the real-time temperature. We also use a PID controller to adjusts the power supplied to the heating elements, ensuring a precise and stable temperature response. It continuously monitors the temperature and makes real-time adjustments to compensate for the factors explained before.

In this attached files it is possible to see the arduino code that controll the machine, and also the PID part.





https://www.instructables.com/F9Y/R2KU/LID5CEA6/F9YR2KULID5CEA6.ino Download

https://www.instructables.com/FSE/SQB6/LID5CEAP/FSESQB6LID5CEAP.h Download