



Electronics for Applied Physics project

Automated grow box for mushrooms using Arduino

WHY?

- The main idea was to create an eco-sustainable package starting by the roots of mushrooms
- After a certain time, we could insert mushrooms in an automated grow box in order to speed up the process



WHY?

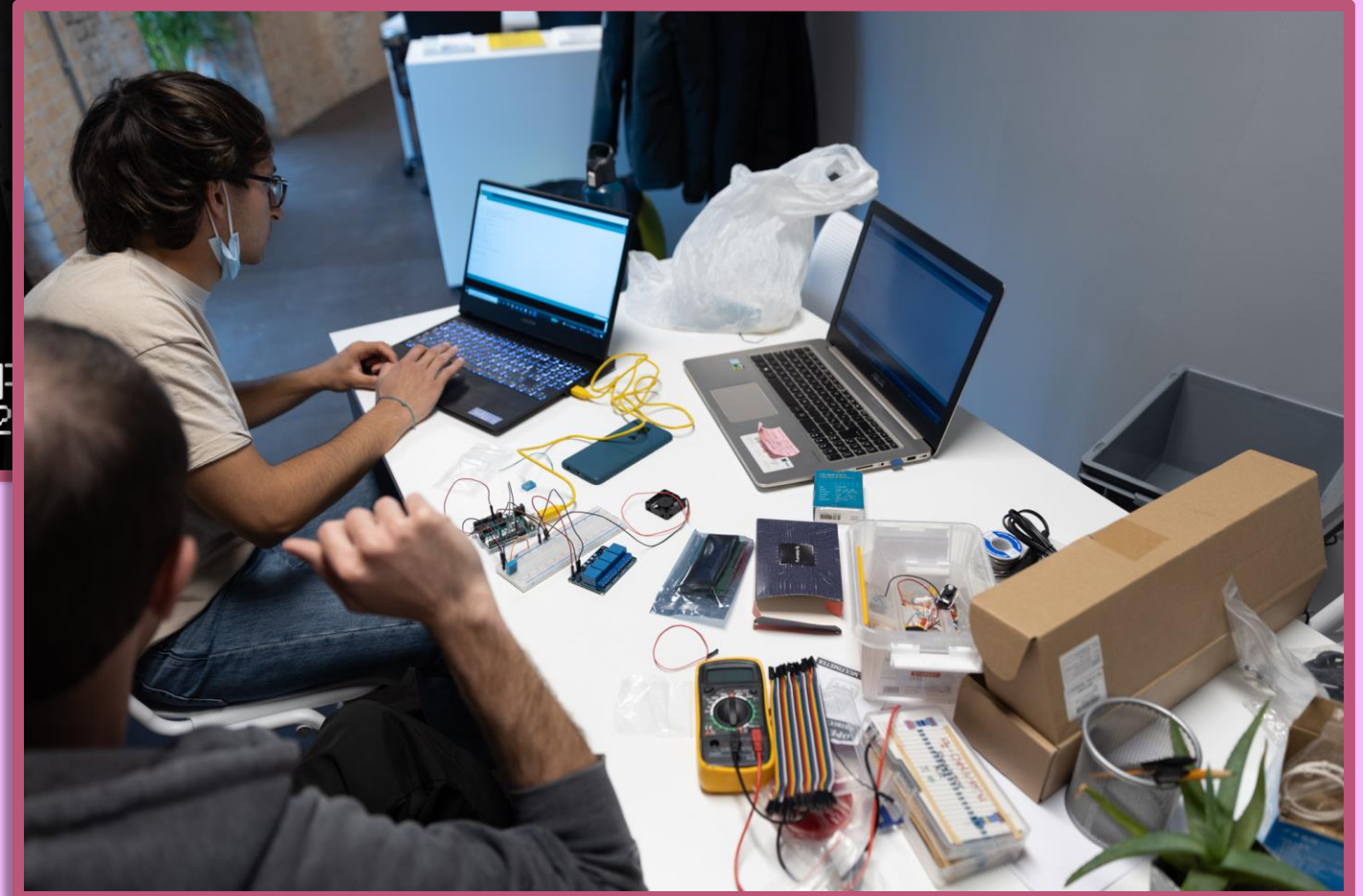
- Mushrooms have been used in order to sponsor *Phylor*, an agency specialized in food supplements by using mushrooms
- The grow box then has been exposed for the **PoliPolo** Fab Lab in Ferrara



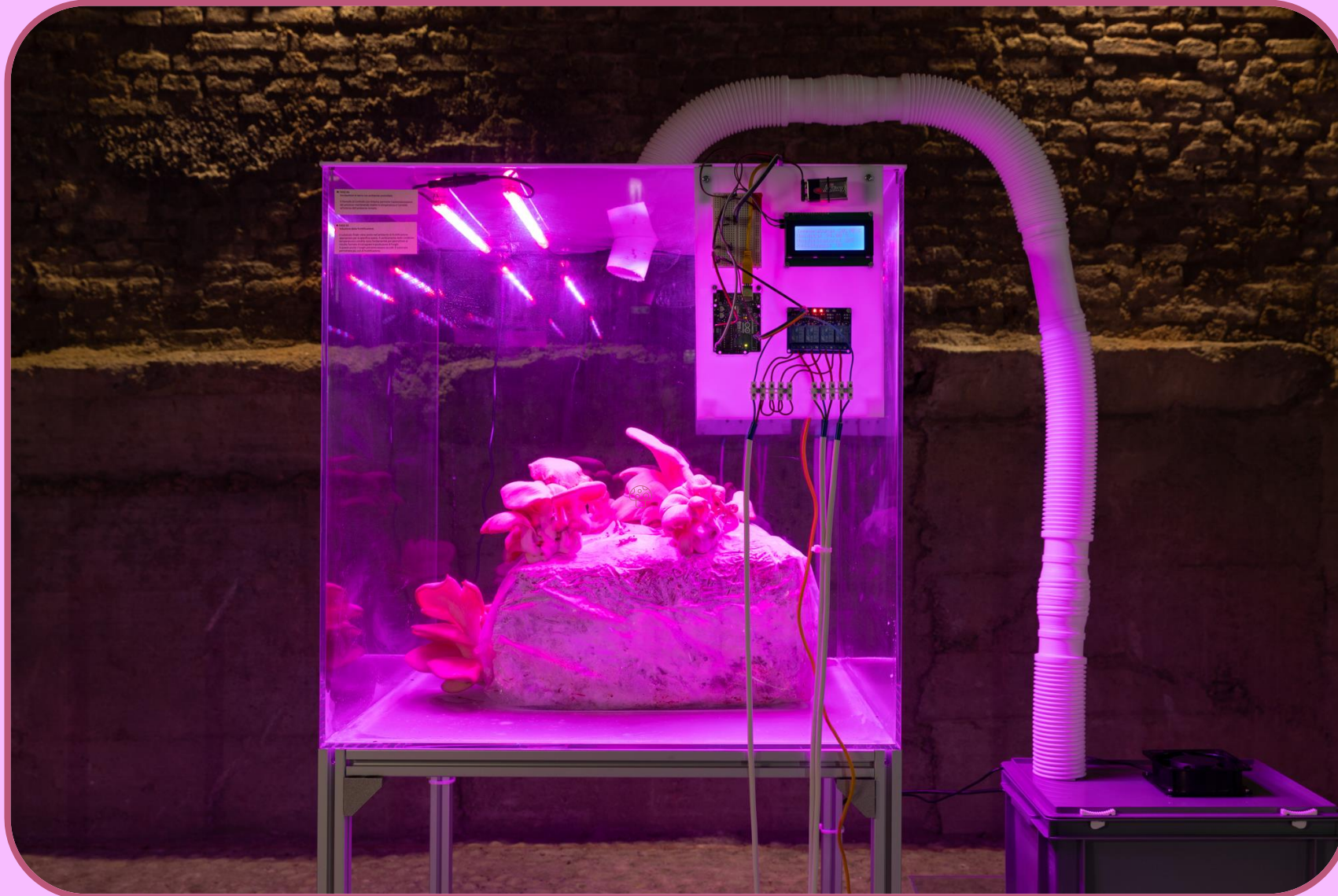
PLAM Creative Studio



gave me the opportunity to work on this project!

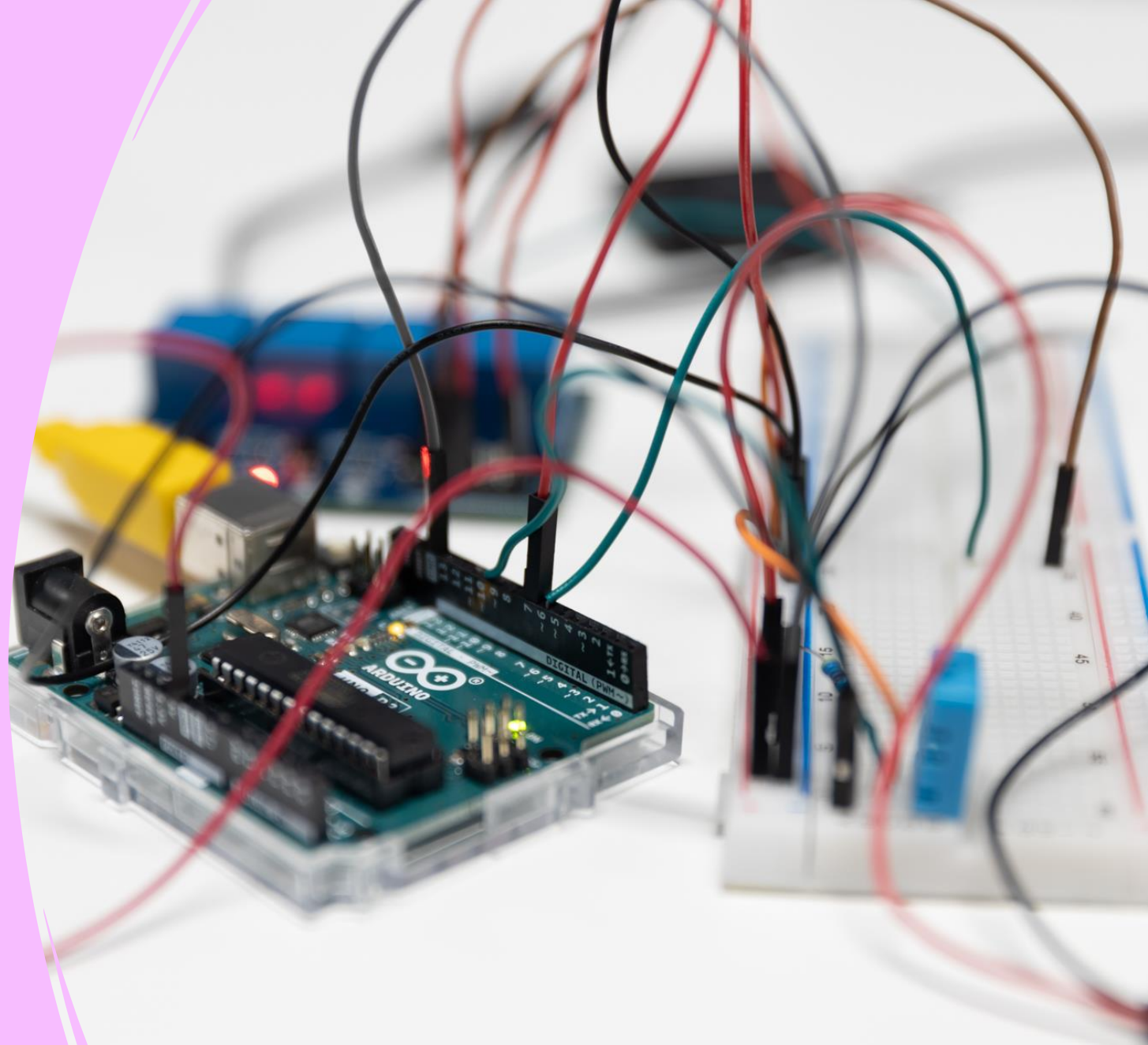


The final result was:



TOOLS REQUIRED

- Arduino UNO
- Breadboard, resistors, jumper wires
- DHT11 sensor
- LCD display
- 4-channel Relay
- 4 led grow lights
- Ultrasonic atomiser
- Fan (optional: potentiometer)
- Heat mat





ARDUINO UNO

- Open-source microcontroller board
- Input Voltage: 7 to 20 Volts
- 14 digital I/O pins (six capable of PWM output)
- 6 analog I/O pins
- 32k of Flash Memory
- Clock speed: 16 MHz
- Programmable with the Arduino IDE

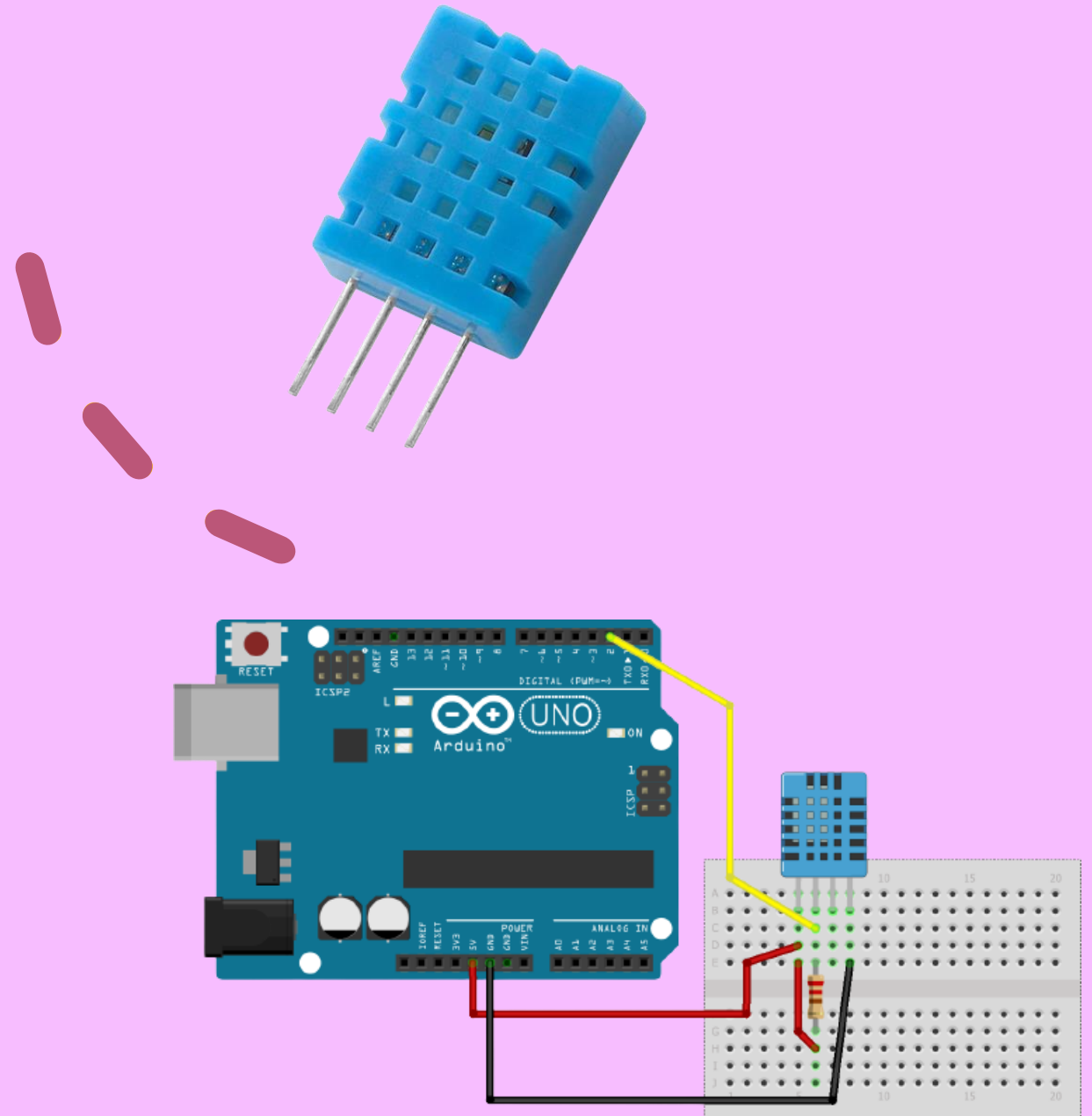


DHT11 – TEMPERATURE AND HUMIDITY SENSOR

- VCC: 3.5V to 5.5V
- DATA: outputs both temperature and humidity through serial data
- NC: no connection (not used)
- GROUND: connected to the ground of the circuit

- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: $\pm 1^\circ\text{C}$ and $\pm 1\%$
- Needs 4,7 K Ω resistor!

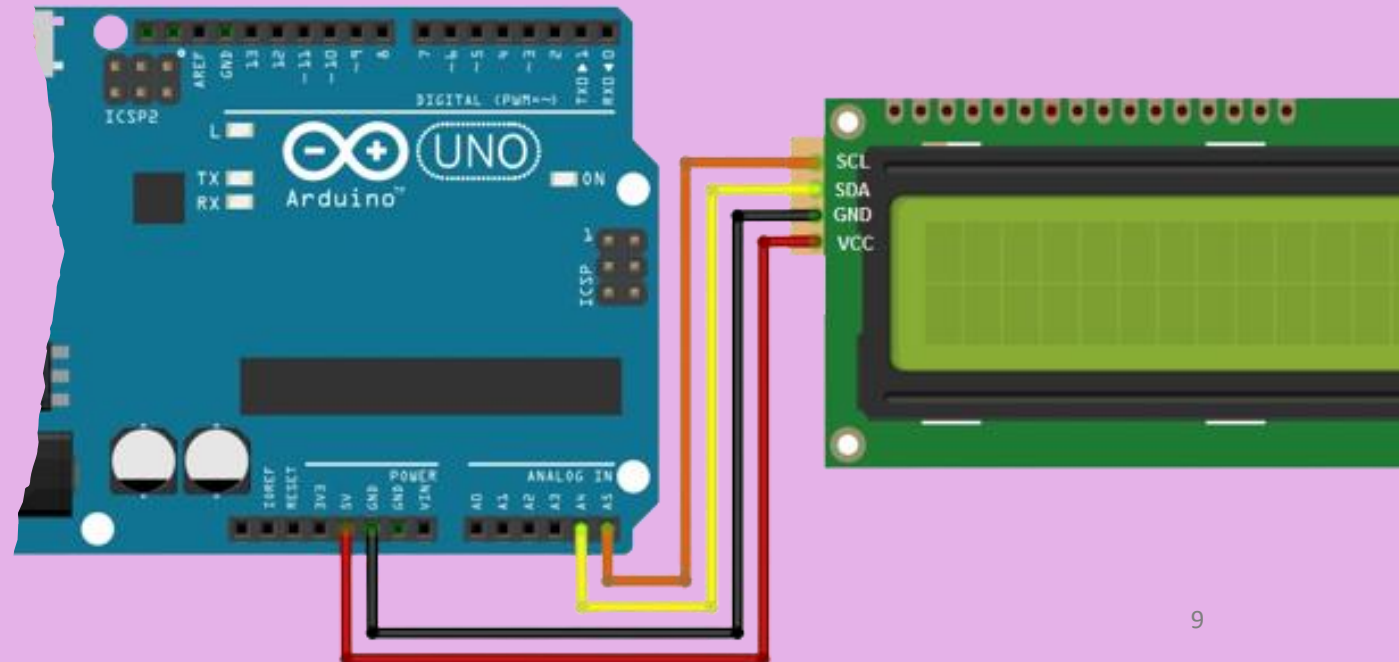
- Implemented with DHT .h library





LIQUID CRYSTAL DISPLAY WITH I2C SERIAL INTERFACE ADAPTER MODULE

- 20x4 character
- 20 total pins
- Just 4 pins connected
- The module has an onboard potentiometer
- Implemented with the `LiquidCrystal_I2C.h` library





4-CHANNEL RELAY

High voltage side

- NO: Normally Open
- COM: Common
- NC: Normally Closed

Low voltage side

- GND
- IN1: Atomiser
- IN2: Fan
- IN3: Light
- IN4: Heat mat
- VCC

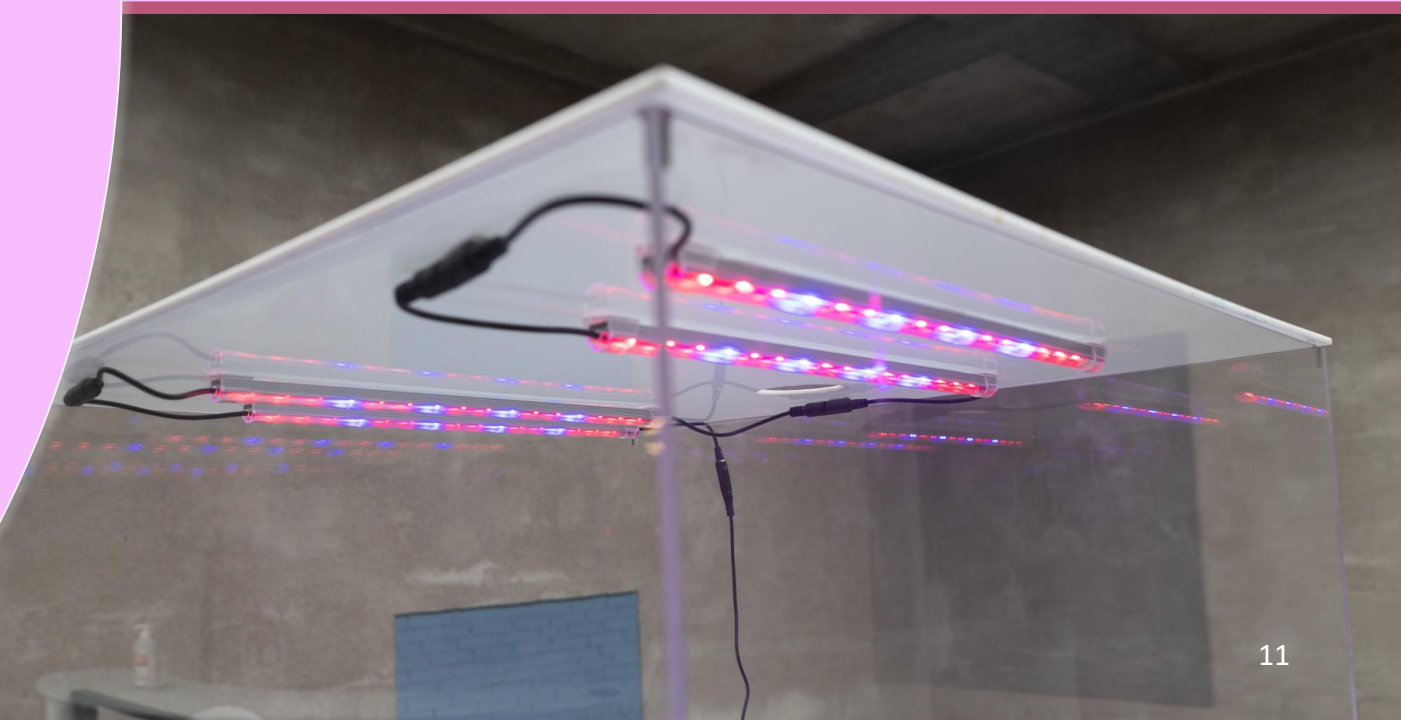
4 LED GROW LIGHTS

- Spectrum given by white light (3000 K), blue (460 nm) and red (620 nm), the mostly absorbed bands by plants
- Input Voltage: 240 V
- Frequency: 50-60 Hz
- Connected to a power strip

12 hours ON and 12 hours OFF using the function `millis()`



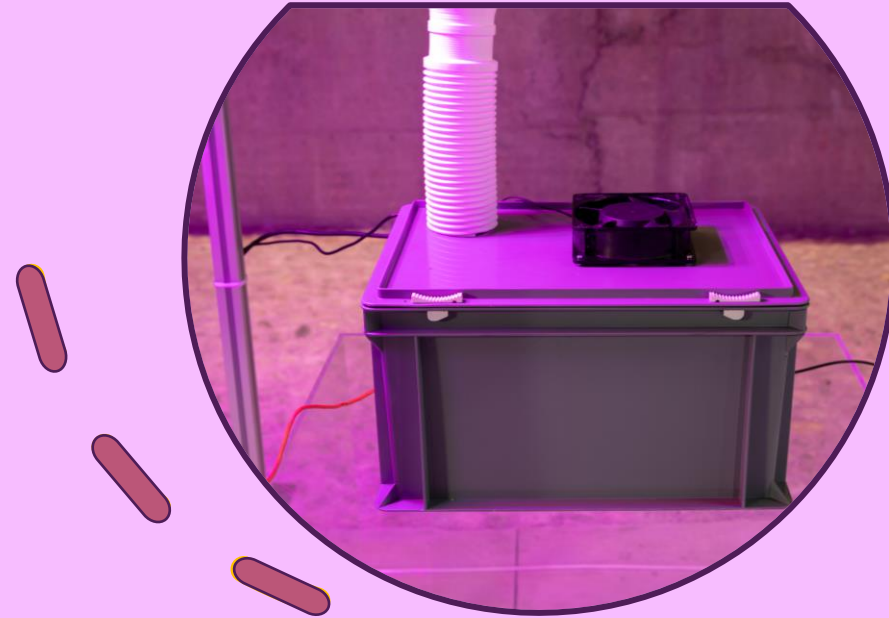
RTC sensor could work better!



ULTRASONIC ATOMISER & FAN

- Input voltage: 24 V
- Water consumption: 80 ml/h
- Ultrasonic pulses break down water molecules
- Connected to a power strip

- Input voltage: 5 V
- Power: 0,6 W
- Flux air: 12 m³/h
- Connected to a power strip

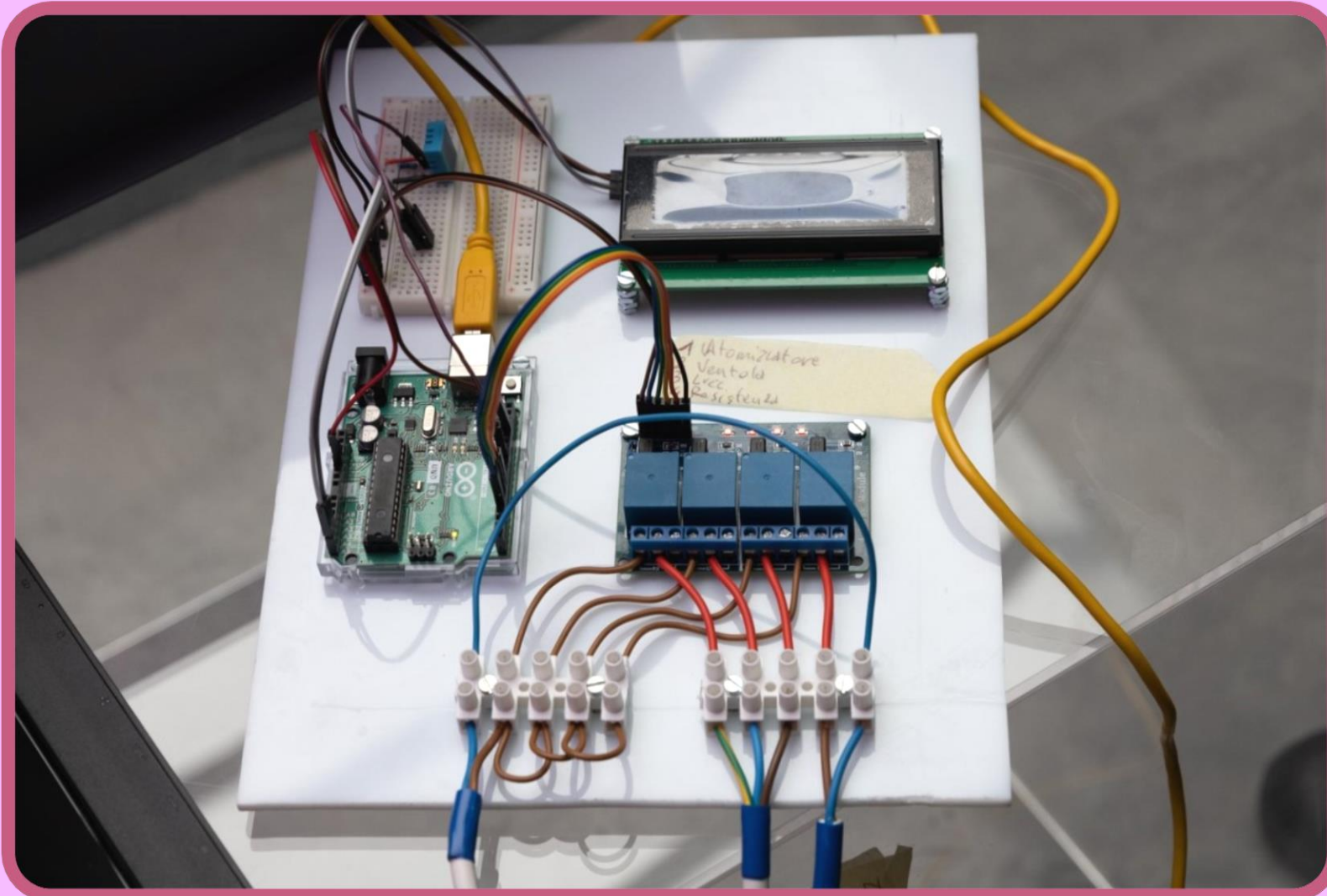




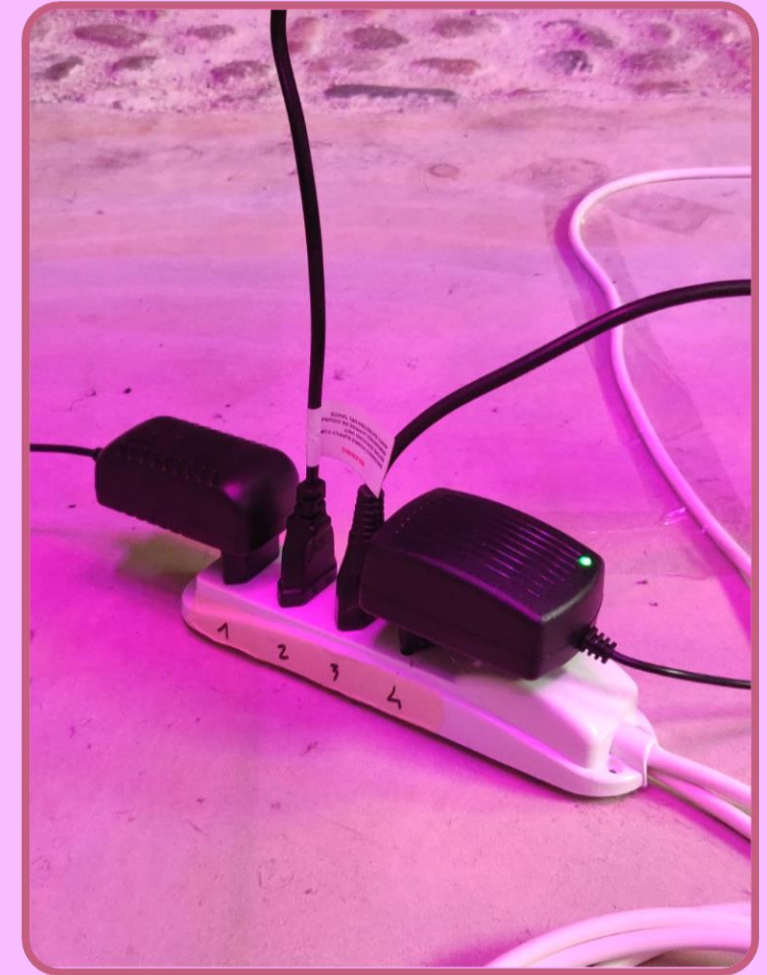
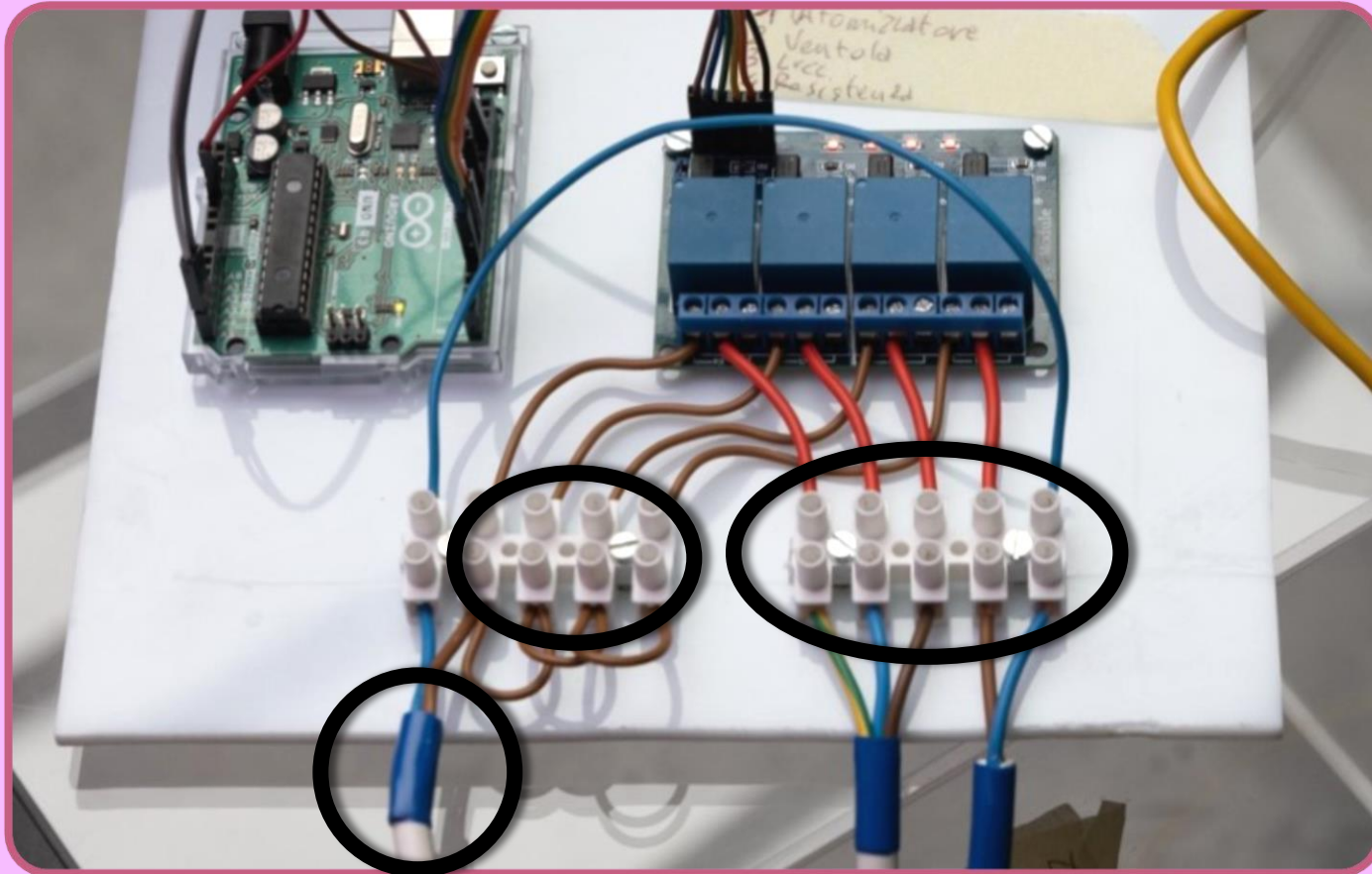
HEAT MAT

- Input Voltage: 220 V
- Power: 20 W
- Keeps the temperature between 20 °C and 42 °C
- Connected to a power strip

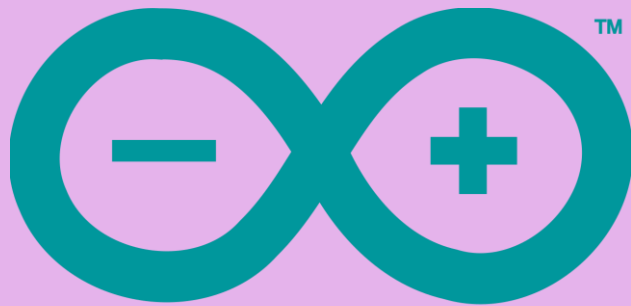
Breadbord, sensor, Arduino, LCD display and relay connected all together



Relay input: 220 V, brown parallel wires represent neutral, right part connected with the power strip



At the end, we can control through Arduino the electrical outlets of the power strip, like a switch!



ARDUINO



SOFTWARE DEVELOPMENT

LOAD LIBRARIES & DEFINITION OF CONSTANTS

```
#include "DHT.h" // library for the humidity-temperature sensor
#include <LiquidCrystal_I2C.h> // library for the LCD Display
LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a
16 chars and 2 line display
```

```
const float minimum_humidity = 80.00; // (%)
const float minimum_temp = 26; // (°C)
```

```
#define DHTPIN 10 // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

```
unsigned long myTime; // for the time management
extern volatile unsigned long timer0_millis;
```




SETUP THE LCD DISPLAY AND THE FIRST MESSAGES

```
void setup() {  
  Serial.begin(9600);  
  Serial.println(F("Let's start!"));  
  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
  
  dht.begin();  
  
  lcd.init();  
  // Print a message to the LCD.  
  lcd.backlight();  
  lcd.clear();  
  lcd.setCursor(0,0);  
  lcd.print("GROW BOX - PoliPolo");  
  lcd.setCursor(0,1);  
  lcd.print("PLAM and Phylor for");  
  lcd.setCursor(0,2);  
  lcd.print("LabAperto Ferrara");  
  lcd.setCursor(0,3);  
  lcd.print("PopUp 29 APR 2022");  
  
}
```

```
void loop() {  
  // Wait a few seconds between  
  measurements  
  delay(2000);  
  
  float humidity = dht.readHumidity();  
  float temp = dht.readTemperature();  
  float f = dht.readTemperature(true);
```

DEFINE VARIABLES AND WRITE THEM ON THE LCD DISPLAY

```
// Print measured temperature  
and humidity on the LCD Display  
  
  lcd.clear();  
  lcd.setCursor(0,0);  
  lcd.print("Temperatura:");  
  lcd.setCursor(13,0);  
  lcd.print(temp);  
  lcd.setCursor(19,0);  
  lcd.print("C");  
  lcd.setCursor(0,1);  
  lcd.print("Umidita:");  
  lcd.setCursor(9,1);  
  lcd.print(humidity);  
  lcd.setCursor(15,1);  
  lcd.print("%");
```

CONTROL THE ELECTRICAL OUTLETS

LED

```
// Set the lights ON for 12 hours,
then turn them off
Serial.print("Time: ");
myTime = millis();

Serial.println(myTime);
// prints time since program
started
digitalWrite(4, LOW);

if (myTime > 43200000) {
    digitalWrite(4, HIGH);
}

// when 24h passed, impose the end
of the loop
if (myTime > 86400000) {
    noInterrupts ();
    timer0_millis = 0;
    interrupts ();
}
```

ATOMIZER & FAN

```
// setting the atomizer and the
fan ON if the humidity is under
the minimum humidity
if (humidity <
minimum_humidity ) {
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    lcd.setCursor(0,2);
    lcd.print("Umidificatore:
ON");
}
else {
    digitalWrite(6, HIGH);
    digitalWrite(7, HIGH);
    lcd.setCursor(0,2);
    lcd.print("Umidificatore:
OFF");
}
```

HEAT MAT

```
// setting the hot resistance
ON if the temperature is under
the minimum temperature
if (temp < minimum_temp ) {
    digitalWrite(5, LOW);
    lcd.setCursor(0,3);
    lcd.print("Resistenza: ON");
}
else {
    digitalWrite(5, HIGH);
    lcd.setCursor(0,3);
    lcd.print("Resistenza:
OFF");
}
```

Normally Open configuration of the relay (NO):

- HIGH signal – current is **not** flowing
- LOW signal – current is flowing

Thanks for your attention!

